COMPUTER PROGRAM DOCUMENTATION

ISOCLS

ITERATIVE SELF-ORGANIZING CLUSTERING PROGRAM

Program C094

By

Ruth T. Minter

Prepared Under Contract NAS 9-12200

By

LOCKHEED ELECTRONICS COMPANY, INC.

HOUSTON AEROSPACE SYSTEMS DIVISION

For

EARTH OBSERVATIONS DIVISION

National Aeronautics and Space Administration

LYNDON B. JOHNSON SPACE CENTER

Houston, Texas

October 1972

Applied Mechanics
Department 628-20
CPD202

COMPUTER PROGRAM DOCUMENTATION

ISOCLS

ITERATIVE SELF-ORGANIZING CLUSTERING PROGRAM

Program C094

by

Ruth T. Minter

Prepared by:

Lockheed Electronics Company, Inc.
Houston Aerospace Systems Division
Houston, Texas

Under Contract NAS 9-12200

For

EARTH OBSERVATIONS DIVISION

National Aeronautics and Space Administration
Johnson Space Center
Houston, Texas

October 1972

MSC PROGRAM NUMBER C094

ITERATIVE SELF-ORGANIZING CLUSTERING PROGRAM (ISOCLS)

Prepared by: Ruth T. Minter
Ruth T. Minter
Scientific Programmer
Lockheed Electronics Company, Inc.

Approved by: WJReicks
W. J. Reicks
Supervisor, Physical Science Section
Lockheed Electronics Company, Inc.

R.T. Theobald
R. T. Theobald
Manager, Applied Mechanics Department
Lockheed Electronics Company, Inc.

J.L. Dragg
J. L. Dragg
Chief/Applications Analysis Branch
Earth Observations Division

T. H. Smith, Jr.
Technical Monitor/Data Requirements Branch
Earth Observations Division

John D. Overton
J. D. Overton
Chief/Data Requirements Branch
Earth Observations Division

National Aeronautics and Space Administration
Johnson Space Center
Houston, Texas
October 1972

TABLE OF CONTENTS

TABLE OF CONTENTS (Concluded)

C094 | COMPUTER PROGRAM ABSTRACT

01 20 TITLE OF PROGRAM (62 CHARACTERS MAXIMUM) Iterative Self-Organizing Clustering Program	01 72 SYMBOLIC NAME (8 CHARACTERS MAXIMUM) ISOCLS	PARENT PROGRAM 02 14 CATEGORY	02 16 SITE	02 19 PROGRAM NO. C094

02 27 LANGUAGE NO. 1 FOR V	02 32 LANGUAGE NO.2 SLEUTH	02 37 KEY WORDS (8 MAXIMUM, SEPARATED BY COMMAS) Clustering, Multispectral Scanner Data, Pattern Recognition, Data Analysis

WHOM TO CONTACT ABOUT THE PROGRAM

05 14 CONTACT Fred Nau	05 28 SITE	05 31 ORGN. CODE	05 39 PROJECT NO. 1659J	05 45 NASA CENTER	05 48 STATUS □ A. UNDER DEVELOPMENT □ B. OPERATIONAL ☒ C. COMPLETED	05 49 □ A. THIS PROGRAM IS NOT FOR SHARING □ B. LIMITED SHARING (SEE ABSTRACT)

DATES 05 50 INITIATED 2-9-72	05 54 COMPLETED 4-25-72	05 58 REVISION CODE ☒ A. REVISION □ B. CANCELLATION	TIME AND COST FOR DEVELOPMENT 05 59 MAN-MONTHS 2.2	05 64 MACHINE HOURS .	05 69 COMPUTER TYPE	05 74 TOTAL COST (DOLLARS) .

CARD NUMBER | COLUMN 14 | ABSTRACT | ELITE MARGIN | PICA MARGIN

This program implements an algorithm which, ideally, sorts a given set of multivariate data points into similar groups or clusters. The program is intended for use in the evaluation of multispectral scanner data, however, the algorithm could be used for other types of data as well.

The user may specify a set of initial estimated cluster means to begin the procedure, or he may begin with the assumption that all the data belongs to one cluster. The procedure is initiated by assigning each data point to the nearest (in absolute distance) cluster mean. If no initial cluster means were input all of the data is assigned to cluster 1. The means and standard deviations are calculated for each cluster.

Each cluster having a standard deviation greater than a given threshold is split into two new clusters. The data is reassigned to the new clusters and means and standard deviations are calculated again. The splitting continues until at least 80 percent of the clusters have standard deviations less than the threshold. Then any clusters whose means are less than a specified distance apart are combined. The data is reassigned and the splitting and combining of clusters alternate for a fixed number of iterations.

Output of the program consists of a line printer map of the clustered data and a summary of statistics for each cluster, and optionally a tape of the clustered data for display on the PMIS DAS.

RELATED DOCUMENTATION (66 CHARACTERS MAXIMUM, SEPARATE EACH REF. BY COMMAS)

ORIGINAL PAGE IS OF POOR QUALITY

INSTRUCTIONS FOR COMPLETING FORM 800

01/7 PROGRAM NUMBER - Program numbers will be assigned upon request by ADP Program Sharing Librarian.

01/14 CATEGORY - A two digit category code assigned by ADP Program Librarian.

01/20 TITLE OF PROGRAM - The title should not contain abbreviations or acronyms.

01/72 SYMBOLIC NAME - Symbolic name (or acronym) program is identified by. Use up to 9 characters. Can be left blank.

02/27 LANGUAGE - Five (5) character abbreviations as follows:

COBOL	=	COBOL
FORTRAN IV	=	FOR4
FORTRAN V	=	FOR5
HYBRID FORTRAN	=	FORHY
SLEUTH II	=	SLEUT

(For other languages, contact the ADP Program Librarian)

02/37 KEY WORDS - A maximum of 8 key words or phrases separated by commas should be used. Do not use abbreviations or acronyms.

05/14 CONTACT - Give last name only, with a maximum of 14 characters.

05/28 SITE - Site where program was developed.

05/45 NASA CENTER - Site responsible for development. To be used by off-site contractors only.

05/49 THIS PROGRAM IS NOT FOR SHARING - If Block 05/49 (A or B) is checked, a statement explaining why program is "limited" or "not" for sharing must be included in the body of the abstract.

05/59 MAN-MONTHS - Enter actual or estimated number of man-months and tenths of man-months required to develop the program.

05/64 MACHINE HOURS - Enter actual or estimated number of computer machine hours and tenths of hours required to develop the program.

A special emphasis should be placed on the selection of the Category, proper choice of Key Words, and wording of Title, since these are the main criteria used in locating programs. Also, the body of the abstract should contain a brief explanation of the program's purpose, functions, restrictions, output, etc. or any other information pertinent in uniquely identifying the program. Avoid the use of abbreviations or acronyms since these terms are often meaningless to someone not closely related to the program.

For additional information contact your center ADP Program Librarian.

1.0 INTRODUCTION

This program performs a modified version of the clustering algorithm known as ISODATA to multispectral scanner data. The acronym ISODATA stands for Iterative Self-Organizing Data Analysis Technique (A). As its name implies, the algorithm is an iterative procedure which groups similar 'objects' into sets called clusters. The algorithm was originally developed by Ball and Hall of Stanford Research Institute and used in their PROMENADE system. (See References 1 and 2 for articles written by Ball and Hall on this subject). A clustering technique based on ISODATA and suitable for MSC's use in processing multispectral scanner data, was developed by E. Kan and A. Holly (LEC). To distinguish between the original and revised programs it was decided to call MSC's version of the clustering program ISOCLS (Iterative Self-Organizing Clustering Program).

The procedure will, ideally, separate all of the data into distinct groups or clusters, the center of each cluster being represented by its mean. The process is initialized by assigning each data point to the nearest estimated cluster center (absolute distance is calculated to each cluster mean). After assigning all of the data to clusters, new means are calculated and tests are made to see if clusters should be split or combined. A cluster is split if the standard deviation of the cluster exceeds a specific threshold value. Two clusters are combined if the distance between the cluster centers is smaller than the specified threshold. A cluster is deleted if it has fewer than some specified number of points. The data is reassigned after each split or combine iteration to the new clusters and the process continues until the desired number of iterations has been obtained.

2.0 PROGRAM DESCRIPTION

2.1 GENERAL DESCRIPTION

The main program ISOCLS is used as a driver to call the major subroutines in the program. The tasks performed by the subroutines called in the main program are summarized below.

SETUP - 1. Reads and analyzes all card input to the program.
2. Initializes default options if not input by user.
3. Does some error checking on input parameters.
4. Calculates the amount of storage needed for the data user has requested and assigns drum storage accordingly.
5. Prints input summary on the line printer.

TAPERD - Reads the users data tape, selects the requested fields and channels from the data, and stores the data unpacked on drum unit 4 if the core storage array IDAT will not hold it all.

ISODAT - Performs the clustering algorithm described in the Technical Description of this document and stores the results on drum unit 7.

COVARR - Calculates and prints the covariance matrix for each cluster.

PCHSTA - Punches means and covariance matrix for each cluster in the same format as LARSYS's STAT processor.

CHAIN - Performs the chaining algorithm described in Section 2.2 of this document.

TWRITE - Creates a tape of the cluster statistics for input to a preprocessor for the Earth Resources Interactive System (ERIPS).

PRINT - Prints the results obtained in the clustering algorithm. This printout includes a summary of the number of points in each cluster, the mean and standard deviation of each cluster, and the distance between cluster centers. In addition, each field is printed (the field is identified by lines and columns), with each data point represented by the symbol assigned to the cluster to which it belongs. PRINT is also called from ISODAT to print cluster summaries and maps.

DASTAP -Creates a tape of the clustered data for display on the PMIS DAS.

2.2 TECHNICAL DESCRIPTION[1]

The aim of any clustering algorithm is to partition a given set of multivariate data points, with little or no knowledge about the actual distribution of the data, into disjoint sets of 'similar' data points. The algorithm implemented in this program uses as a measure of similarity the absolute distance between a data point and the cluster mean. The procedure begins with, optionally, a set of assumed cluster means or the assumption that all the data is one cluster and proceeds with a series of "split" and "combine" iterations, until the maximum number of iterations is reached.

[1] This technical description is based on information contained in documents written by E. Kan (References 3-6). Excerpts are lifted from the documents in some instances. An attempt was made to keep notations the same, however, where symbolic names are used, the actual Fortran name for the variable in the program is used in this discussion.

The algorithm is detailed below step by step following the notation definitions.

NOTATION DEFINITIONS

SYMBOL	FORTRAN NAME(S)	DEFINITION
CLD_{ij}	CLD(I,J)	Intercluster distance between clusters I and J.
$d(X_k,\mu^{(I)})$	DIST	Distance from the data point k to the center of cluster I.
DLMIN	DLMIN	Threshold value for combining clusters.
ISTOP	ISTOP	Maximum number of iterations.
LNCAT	LNCAT INCAT	Number of existing clusters at a given time.

N(I)	N(I)
	DN(I)
		Total number of data points assigned to cluster I.

NMIN	NMIN	Minimum number of data points allowed per cluster.
NOFEAT	NOFEAT	Number of coordinates in a data vector.
STDMAX	STDMAX	Threshold for splitting clusters.
X_k	C(J,K)	Data vector k, $C(I,K)=(X_{1k},X_{2k}$ . . . $X_{NOFEAT\ k})$

$\mu_j^{(I)}$	MEANS(J,I)
	AMN(J,I)
		Mean of the j^{th} coordinate of the I^{th} cluster.

$\gamma_j^{(I)}$	AVP(J,I)	Temporary summing variable for the calculation of the standard deviation of the j^{th} coordinate of the I^{th} cluster.

SYMBOL	FORTRAN NAME(S)	DEFINITION
$\sigma_j^{(I)}$	STDEV(J,I)	Standard deviation of the j^{th} coordinate of the I^{th} cluster.

INITIALIZE

- Initialize threshold values for splitting clusters (STDMAX), combining clusters (DLMIN) and deleting clusters (NMIN).

CLASSIFY AND CALCULATE NEW STATISTICS

- Assign each data point to a cluster and at the same time collect the means, standard deviations and point count of the newly developing clusters.

1. Assign the data point $X_k = (X_{k1}, X_{k2} \ . \ . \ . \ X_{k\ NOFEAT})$ to the I^{th} cluster if $d(X_k, \mu^{(I)}) \leq d(X_k, \mu^{(J)})$ for all $J \neq I$, where $d(X_k, \mu^{(I)})$ is defined as

$$d(X_k, \mu^{(I)}) = \sum_{j=1}^{NOFEAT} \left| X_{kj} - \mu_j^{(I)} \right|$$

2. $N(I) = N(I) + 1$

3. $\mu_j^{(I)} = \frac{N(I)-1}{N(I)} \mu_j^{(I)} + \frac{1}{N(I)} X_{kj}$

4. $\gamma_j^{(I)} = \frac{N(I)-1}{N(I)} \gamma_j^{(I)} + \frac{1}{N(I)} X_{kj}^2$

5. $\sigma_j^{(I)} = \left\{ \gamma_j^{(I)} - \left(\mu_j^{(I)} \right)^2 \right\}^{1/2} \qquad j = 1, \ldots NOFEAT$

Return to 1. until all data points have been classified.

DELETE

Delete all clusters which have fewer than NMIN members. A cluster is deleted simply by removing the statistics for that cluster and reducing the number of clusters (LNCAT) accordingly.

DETERMINE TYPE OF ITERATION

Determine whether this is to be a SPLIT iteration or a COMBINE iteration and proceed to appropriate step. The sequence of iterations will be as follows.

$$\underbrace{\underbrace{\mathrm{SSSS}}_{n}\mathrm{CSCSC....S}}_{\mathrm{ISTOP}} \qquad \text{where} \quad \begin{array}{l} \mathrm{S} = \text{Split Iteration} \\ \mathrm{C} = \text{Combine Iteration} \end{array}$$

The beginning sequence of split iterations is terminated when at least 80% of the clusters have standard deviations less than the threshold parameter STDMAX. At that point the iterations alternate between combine and split until the last (ISTOP) iteration, which is always a split iteration.

The initial split iterations are for the automatic initialization of cluster centers in the event they are not input. The sequence is shortened considerably if initial cluster centers are input.

SPLIT

A cluster is split along the j^{th} coordinate if (1) the j^{th} coordinate has the maximum standard deviation for the cluster, (2) the standard deviation along the j^{th} coordinate is greater than the threshold STDMAX; and if (3) the cluster has more than 2 (NMIN + 1) data points.

If the above conditions are met then two new clusters are created and the parent cluster is deleted. A cluster is created merely by defining its centers (means) for each coordinate. If the I^{th} cluster is split in the j^{th} coordinate then the two new clusters will have centers at

$(\mu_1^{(I)}, \mu_2^{(I)}, \ldots \mu_j^{(I)} \pm \alpha, \ldots \mu_{NOFEAT}^{(I)})$. Where α will normally be $\sigma_j^{(I)}$, but can be a constant input by user (See Card Input Section 3.1.1, SEP control card). On a given split iteration all clusters having a standard deviation greater than STDMAX will be split provided the maximum number of clusters has not been reached. In which event reclassification of the data will continue without the creation of new clusters.

COMBINE

Two clusters are combined if the distance between them is less than the threshold parameter DLMIN. The distance between clusters i and j is calculated as

$$CLD_{ij} = \left(\sum_{k=1}^{NOFEAT} (\mu_{ki} - \mu_{kj})^2 / \alpha_{ki}\alpha_{kj} \right)^{1/2}$$

If $CLD_{ij} <$ DLMIN and $CLD_{ij} < CLD_{im}$ for all $m \neq j$ and $m > i$, then the clusters I and J will be merged to form a new cluster L with means

$$\mu_k^{(L)} = \frac{N(I)\mu_k^{(I)} + N(J)\mu_k^{(J)}}{N(I) + N(J)} \qquad k = 1, \text{NOFEAT}$$

The clusters I and J are deleted. The new cluster L is not considered as a candidate for merging with any other cluster on the iteration in which it was formed.

BEGIN NEXT ITERATION

If the maximum number of iterations (ISTOP) has not been reached begin the next iteration with "CLASSIFY AND CALCULATE NEW STATISTICS". If the maximum number of iterations has been reached then reclassify and calculate statistics in case new clusters were formed or deleted on the last iteration, and then proceed to the chaining algorithm.

CHAIN

The last step in the clustering procedure groups all clusters which have intercluster distances less than the chaining threshold (CHNTHS) to form one cluster. The chaining procedure was adopted because the minimum variance criteria used in the iterative procedure above tends to group the data into spherical (or ellipsoidal) groupings with Gaussian distributions. This type of grouping is certainly a natural grouping and would quite often be completely satisfactory. However, there could be natural groupings of the data which are odd shaped and can not be approximated by Gaussian distributions. Two examples are given in Figure 1. At the end of the sequence of split and combine iterations groupings of the type in Figure 1 are likely to be separated into subclusters as illustrated in Figure 2. The chaining algorithm will group the subclusters 1, 2, and 3 (Figure 2) into one composite cluster, likewise clusters 4, 5, 6, and 7 would be grouped together to form one cluster.

The algorithm scans the intercluster distance table (CLD) and begins a chain with the first appearance of two clusters within a distance of CHNTHS units. Once a subcluster is in the chain all clusters which are within CHNTHS units of the subcluster are added to the chain. See Example Figure 3.

The statistics (means, standard deviations and covariance matrices) of the clusters resulting from chaining are not calculated by the program because often the chained cluster cannot be represented by a Gaussian distribution.

There are, of course, instances that one can safely combine into one composite (Gaussian) cluster those subclusters that are chained by the program. For example, the three subclusters 1, 2, and 3 in Fig. 4 can safely be combined into one final cluster. An indication of such possibility would be the fact that these three subclusters are all pairwise close to one another. In this case, the following formulas [3] can be used iteratively to compute the composite statistics:

Assuming that two clusters $\{n_1, m_1, C_1\}$ and $\{n_2, m_2, C_2\}$ are to be considered as one cluster n, m, C , where n's, m's, C's are respectively the number of points, mean vectors and covariance matrices. Then

$$n = n_1 + n_2$$

$$m = \left(\frac{n_1}{n_1 + n_2}\right) m_1 + \left(\frac{n_2}{n_1 + n_2}\right) m_2$$

$$C = \left(\frac{n_1}{n_1 + n_2}\right) C_1 + \left(\frac{n_2}{n_1 + n_2}\right) C_2 + \left(\frac{n_1}{n_1 + n_2}\right) m_1 m_1^T + \left(\frac{n_2}{n_1 + n_2}\right) m_2 m_2^T - mm^T$$

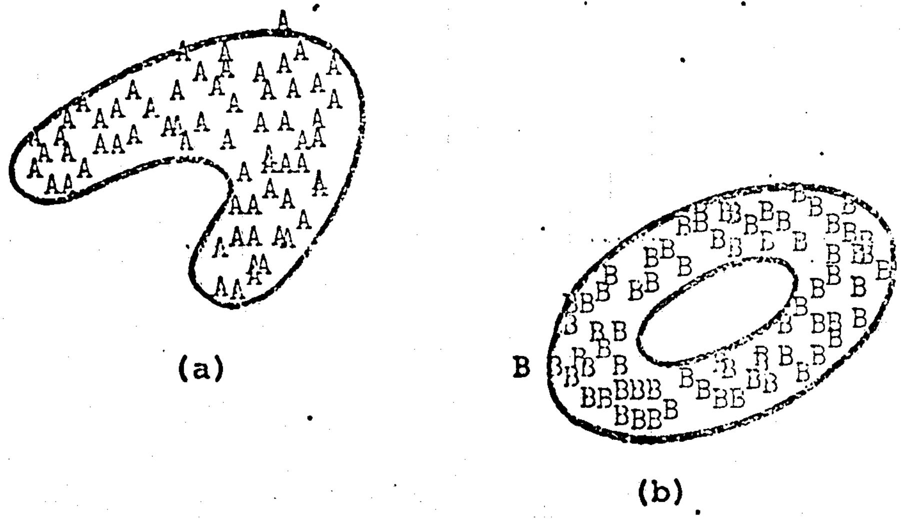


FIG. 1

(a) THE BOOMERANG - SHAPED CLUSTER

(b) THE DONUT - SHAPED CLUSTER

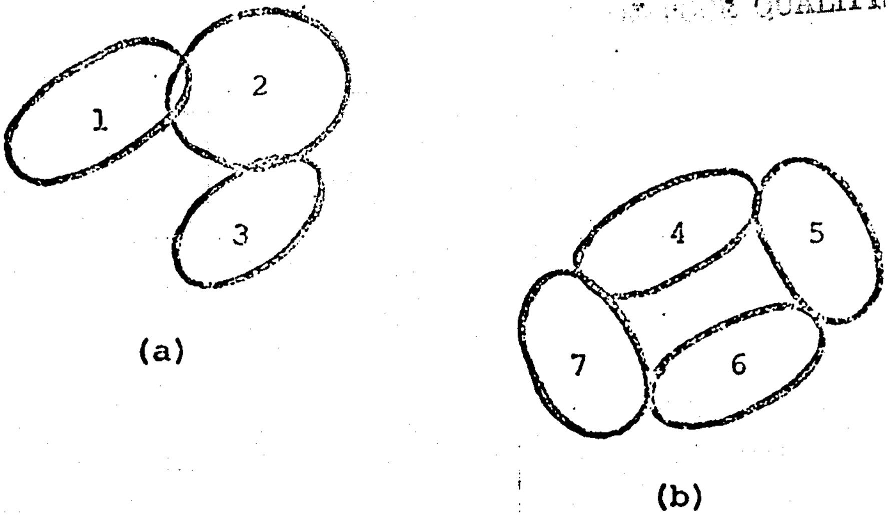
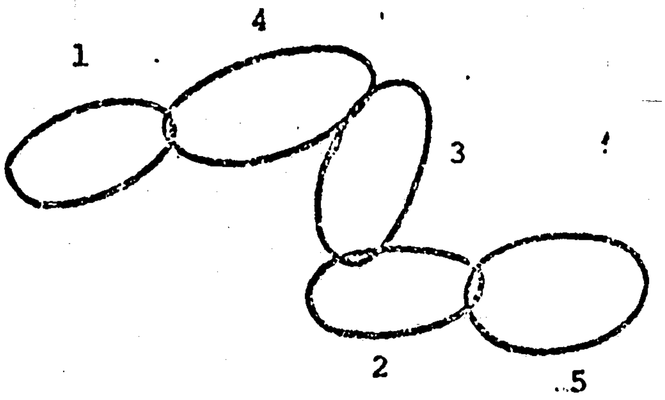


FIG. 2

BREAKING UP OF THE CLUSTERS (a) AND (b)
OF FIG. 1 INTO SUBCLUSTERS

(a)



(b)

J \ I	1	2	3	4	5
1	0.0	7.5	6.2	3.2	11.8
2	7.5	0.0	3.1	5.6	3.0
3	6.2	3.1	0.0	3.1	6.3
4	3.2	5.6	3.1	0.0	9.7
5	11.8	3.0	6.3	9.7	0.0

CHNTHSD = 3.2

FIG. 3. – EXAMPLE FOR CHAINING

(a) CLUSTER STRUCTURE

(b) INTERCLUSTER DISTANCE TABLE

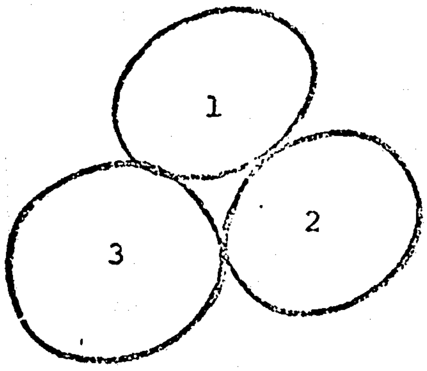


FIG. 4

AN EXAMPLE IN WHICH THE CHAINED SUBCLUSTERS CAN SAFELY BE COMBINED INTO ONE COMPOSITE CLUSTER

A few words of caution are at hand as to the values of DLMIN and STDMAX in the COMBINE and SPLIT routines. The range of values 3.2-3.9 for DLMIN have been established in connection with the probability of misclassification. Values outside this range are discouraged. Of course, values of DLMIN closer to the lower bound will induce finer groupings than otherwise. As to the value of STDMAX, its value directly governs the size of nominal sized clusters. For agricultural data collected by aircraft multispectral scanners having a range of levels from 0 to 255, a value 4.5 of STDMAX is suggested. Higher values of this threshold is acceptable, e.g., 6.0 or 7.0, inducing coarser groupings. For ERTS data with levels from 0 to 127, probably (since there is no experience yet) a value of 2.5 to 4.0 would be suitable.

PRECEDING PAGE BLANK NOT FILMED

3.0 PROGRAM USAGE

3.1 INPUT DESCRIPTION

The user must provide two types of input to the program, (1) card input specifying various optional input parameters, initial cluster means if any,and boundaries of fields to be clustered, (2) tape input containing the multispectral scanner data to be clustered.

3.1.1 Card Input

There are three different formats for the card input to the program.

- Control cards which specify various input options requested by the user. These cards are identified by a key-word left justified in columns 1-10 of the card, with the parameter value(s) in columns 11-80 (beginning in any column past 10). These control cards may be in any order but they should be the first cards in the input deck. The following is a list of all available options in the program along with their default value if provided.

Keyword	Parameter Value	Function
FEATURES	C1, C2, C3 ... (Default - none)	Use the channels indicated in the parameter list for the clustering procedure. C1, C2, C3, ..., should be integer numbers separated by commas.
ERIPS	N (Default-no ERIPS interface tape)	Create a tape for the ERIPS preprocessor. A save tape has been assigned to unit N.

Keyword	Parameter Value	Function
SYMBOLS	S1, S2, S3, ... Default (1,2...9 A, B...Z,%,#,Δ,/,-, *,+,$,@,=,&,?,',., blank	Use the symbols indicated in the parameter list to identify clusters in the printout.
ISTOP	N (Default = 10)	Perform N iterations in the clustering procedure and stop.
LNCAT	N (Default = 1)	The number of initial estimated clusters.
NMIN	N (Default = 30)	Delete any cluster with fewer than N members.
DLMIN	X (Default = 3.2)	Combine any two clusters whose means are closer than X.
SEP	X (Default - maximum of the channel's standard deviations in the cluster)	Upon splitting a cluster, separate the new clusters by a distance of X.
STDMAX	X (Default = 4.5)	Split any cluster whose maximum standard deviation is greater than X.
MAXCLS	N (Default = MAXPOP)	Maximum number of clusters N must be less than or equal to the parameter variable MAXPOP (which is set at 50).

Keyword	Parameter Value	Function
KRN	N (Default = 1)	Print out a summary of the clusters at every Nth iteration.
MAP	N (Default = 20)	Print out a map of the clustered data along with the summary every Nth iteration.
FORMAT	N (Default = 2)	N=1, indicates the input data tape is in the UNIVERSAL format. N=2, indicates the input data tape is in the LARSYS II format.

PUNCH	N
	(Default - no cards punched)
		Punch the means and covariance matrix for each cluster.
		N = 1, format for 1108 programs
		N = 2, format for 360/75 programs
		N>2 or blank, both formats.

DASUNIT	N (Default - no DAS output tape)	Create a tape for the DAS. A save tape is assigned to unit N. N cannot be 3, 4, or 7 since these units are used internally by the program.
HEDI	Any 60 characters (Default - standard heading)	Replace the first line of the standard header record with the indicated 60 characters.
HED2	Any 60 characters (Default - standard heading)	Replace the second line of the standard header record with the indicated 60 characters.
CHAIN	X (Default - chaining not performed)	Chain all clusters within X units of each other to form one cluster.

Keyword	Parameter Value	Function
COMMENT	Any 60 characters (Default - no comment)	Print the indicated comment along with the header.
DATE	Any 12 characters (Default - present date)	Print the date indicated in the header.
END	(Blank)	Indicates the end of one set of control cards.

The remaining two types of input cards follow this card in the deck set-up.

$END*	Blank	Indicates the end of all card input for one data set.

- Means of Initial Clusters

  Estimated means of initial clusters is an optional input. Inputting initial means should decrease the number of iterations required to cluster the data. If this option is exercised, the control card LNCAT must be set to the number of initial clusters.

  Estimated means must be input for each channel listed on the FEATURES card, one value for each channel for each initial cluster. The means immediately follow the *END* card listed above, they are read with an 8F10.2 format. Punched 8 values per card with a 10 column field width, the means should be ordered all channels (in same order as FEATURE card) for the first cluster, followed by all channels for the second cluster, and so on for all initial clusters.

- Field Boundary Definitions

The fields to be clustered are defined by beginning and ending scan line with an increment, and beginning and ending sample point with an increment. These cards immediately follow the means or *END* if no means input. All fields are clustered together as one data set, but the boundary lines and samples are maintained for printout purposes. The format for these cards is as follows.

Columns	Format	Input Definition
1-6	A6	Any six character name to identify this field. (May be blank).
21-25	I5	First line number
26-30	I5	Last line number
31-35	I5	Line increment
36-40	I5	First sample number
41-45	I5	Last sample number
46-50	I5	Sample increment

**Integer formats must be right-justified in the column field.*

All card input to the program is read and analyzed by subroutine SETUP.

3.1.2 Tape Input

The input tape contains the data to be clustered. It must be in either the UNIVERSAL format or the LARSYS II format, and it must be assigned to Fortran unit 3 or logical unit C. The LARSYS II format is defined in Appendix B. The UNIVERSAL format is defined in Reference 7.

Subroutine TAPERD reads the data tape, unpacks the data and selects what the user has requested. The requested data is stored on drum unit 4, unpacked, for use in the clustering procedure.

3.2 PROGRAM RUN PREPARATIONS

The deck set-up for execution on the UNIVAC 1108 is given on page 3.8. Sample for Deck Setup and input is given in Section 5.5.1.

3.3 OUTPUT DESCRIPTION

3.3.1 Line Printer Output

Line printer output consists of the following items:

1. Input summary printed by subroutine SETUP.
2. Summary of the cluster structures and map of the data printed by subroutine PRINT. This includes means and standards deviations of each cluster and count of points assigned to each cluster.
3. According to the value of KRN, messages are printed by subroutine ISODAT when clusters are split, combined or deleted.
4. Covariance matrix of each cluster printed by subroutine COVARR.
5. Summary of chaining algorithm printed by subroutine CHAIN.
6. Diagnostic messages printed in several routines. The diagnostic messages are listed in Appendix A with explanations and recommendations for user action.

Sample Output is given in Section 5.5.2.

3.3.2 Card Output

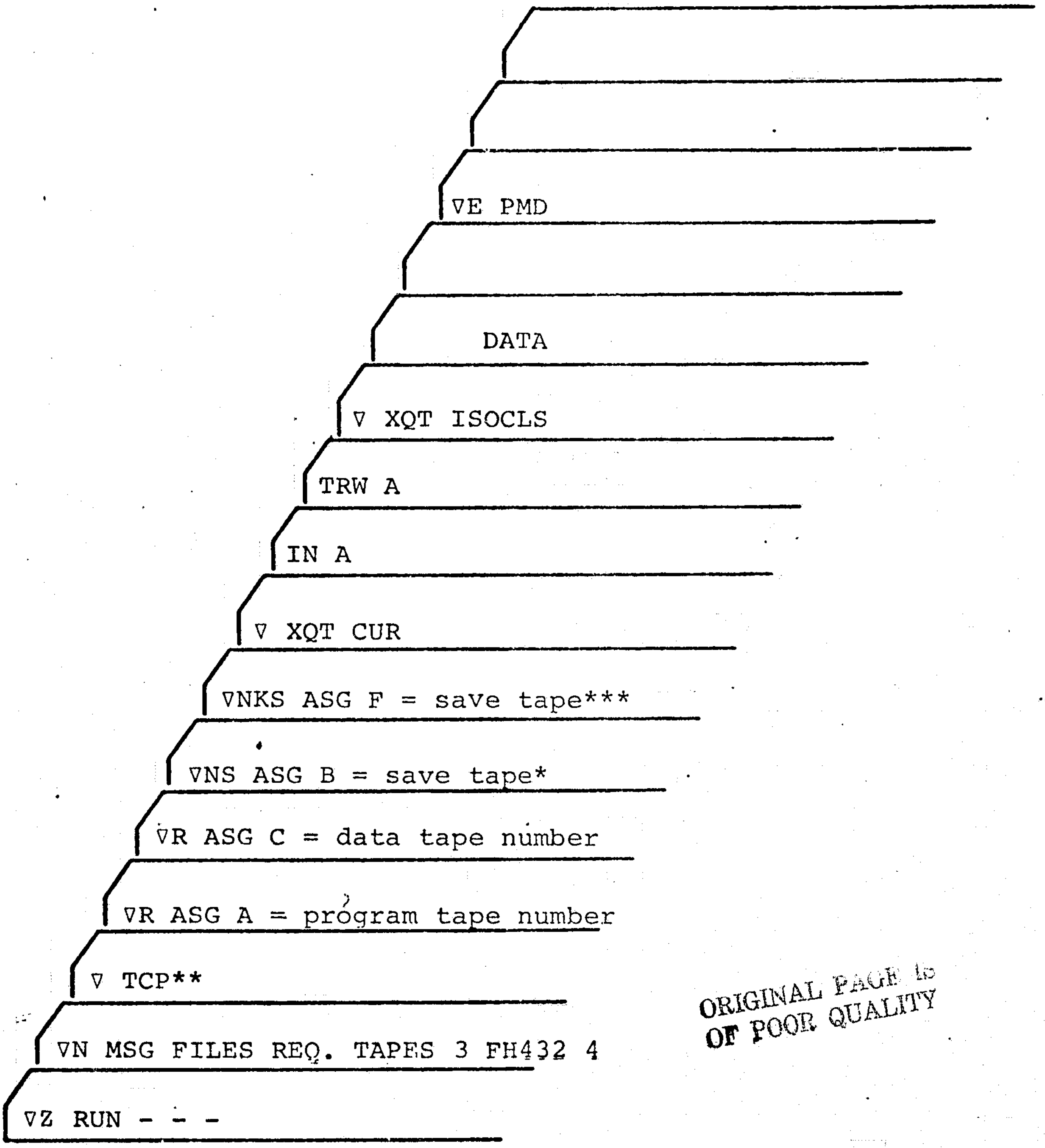
The subroutine PCHSTA outputs on cards the statistics (means and covariance matrix) for each cluster.

Punched output is controlled by the input card PUNCH (See Section 3.1.1). The statistics can be punched in either or both of two formats. One format is acceptable to the 1108 versions of the LARSYS and TLU programs. The other format is acceptable to the 360/75 ERIPS system.

The punched card output does not reflect the chaining algorithm (i.e., the cards are punched for all subclusters before chaining).

DECK SETUP FOR PROGRAM ISOCLS PAGE NO. 1 OF 1

(Back of deck)



```
∇E PMD

DATA

∇ XQT ISOCLS
TRW A
IN A
∇ XQT CUR
∇NKS ASG F = save tape***
∇NS ASG B = save tape*
∇R ASG C = data tape number
∇R ASG A = program tape number
∇ TCP**
∇N MSG FILES REQ. TAPES 3 FH432 4
∇Z RUN - - -
```

(Front of deck)

* Omit this card if the DAS tape option is not used.

** Omit this card if the PUNCH option is not used.

*** Omit this card if the ERIPS tape option is not used.

3.3.2 Tape Output

Subroutine DASTAP creates a data tape which will allow the user to display the results of the clustering on the PMIS DAS. The format of this tape is as follows:

(a) The number of files on the tape equals the number of fields input by the user.

(b) The number of records on each file equals the number of scan lines in the particular field.

(c) Each record is a series of 16 bit bytes, each byte containing the number of the cluster to which the corresponding data point belongs.

This tape should be mounted on a 9-track tape drive so that it will be compatible with the PMIS DAS.

Subroutine TWRITE creates a data tape containing the statistics of the clusters. This tape is an interface between the ISOCLS program and a preprocessor for the Earth Resources Interactive Processing System (ERIPS). It should be mounted on a 9-track drive, using the BCD hardware converter so that it will be compatible with the ERIPS preprocessor.

4.0 EXECUTION CHARACTERISTICS

4.1 Restrictions

Since the program uses drum for internal storage of the data, the user is somewhat limited in the amount of data he can cluster at one time. The program is designed to use drum storage only if the amount of data requested by the user exceeds internal dimensions of the program. The number of data points times the number of channels cannot exceed the number of words available on the FH432 drum. Presently there are 786,432 words of storage available on the FH432 drum, however this number may change with the system. The program prints a diagnostic message if the user has requested too much data. See subroutine SETUP (5.3) for a description on how the drum units are assigned.

Dimensions in the program restrict the number of clusters to fifty, the number of features to thirty and the number of fields to one hundred. The program was designed so that these dimensions could be easily changed if necessary. See Section 5.2 for instruction on how to change these dimensions.

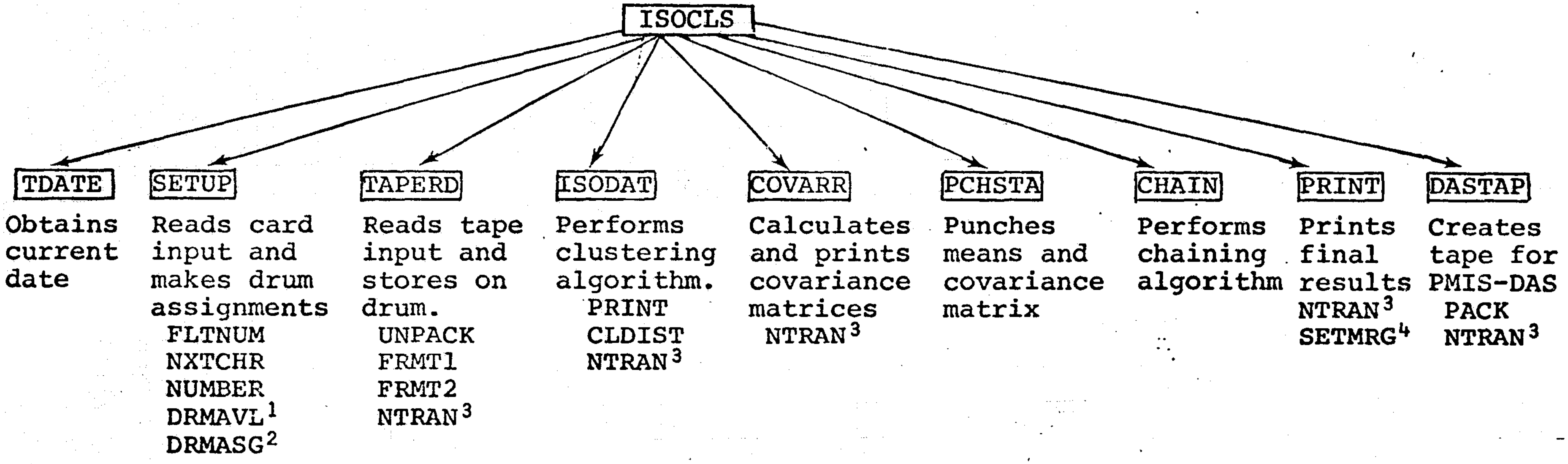
4.2 Running Time/Lines of Output

Run time for the program depends on several factors (1) the number of data points being clustered, (2) the number of channels requested, (3) the number of iterations and (4) the number of times printout is requested. Execution time for a sample case where 1524 data points from 3 channels were clustered, iterating 30 times and printing every third iteration, was 2 minutes and 42 seconds. Increasing the number of data points to 7882 with the other variables the same increased execution time to 6 minutes and 20 seconds.

Lines of printed output will depend on how many iterations are printed. This is controlled by the user with the input parameters KRN and MAP (see Input Description). Printing every iteration for 30 iterations should not exceed 300 pages.

5.0 REFERENCE INFORMATION

5.1 General Flow of the Program


ISOCLS
TDATE
Obtains current date
SETUP
Reads card input and makes drum assignments
FLTNUM
NXTCHR
NUMBER
DRMAVL[1]
DRMASG[2]
TAPERD
Reads tape input and stores on drum.
UNPACK
FRMT1
FRMT2
NTRAN[3]
ISODAT
Performs clustering algorithm.
PRINT
CLDIST
NTRAN[3]
COVARR
Calculates and prints covariance matrices
NTRAN[3]
PCHSTA
Punches means and covariance matrix
CHAIN
Performs chaining algorithm
PRINT
Prints final results
NTRAN[3]
SETMRG[4]
DASTAP
Creates tape for PMIS-DAS
PACK
NTRAN[3]

[1] System routine which returns the number of words available on FH432 and FH1782 drums.

[2] System routine which assigns specific word length to drum unit.

[3] System binary I/O routine.

[4] System routine which sets the margin at the top and bottom of the page.

5.2 'COMMON' Symbol Definitions

The procedure element COMMON is a list of FORTRAN statements which are included in most of the subroutines throughout the program. They are defined in a procedure so that changes may be made easily. Changing a statement in the procedure results in the changes being made throughout the program. However when a change is made to the procedure, all routines including the procedure must be recompiled.

The common block PASS is used to pass variables from one subroutine to another. Symbol definitions for this common block are given below.

FORTRAN Name	Mode	Definition	ROUTINE WHERE INITIALIZED OR CALCULATED
HEAD	A	Array containing the heading to be printed at the top of each page of output.	MAIN PROGRAM or SETUP
NOFEAT	I	Number of channels requested by the user on the input card FEATURES.	SETUP
FETVEC	I	Array containing the specific channels requested by the user.	SETUP
ISTOP	I	Maximum number of iterations for the clustering procedure.	SETUP
LNCAT	I	Number of clusters at a given time.	SETUP & ISODAT

FORTRAN Name	Mode	Definition	ROUTINE WHERE INITIALIZED OR CALCULATED
NMIN	I	Minimum number of points to allow per cluster.	SETUP
KRN	I	Print summary of clusters every KRNth iteration.	SETUP
STDMAX	R	Maximum standard deviation allowed before splitting a cluster.	SETUP
DLMIN	R	Minimum distance between clusters before combining.	SETUP
SEP	R	Distance to separate clusters upon splitting.	SETUP or ISODAT
MEANS	R	Means (one for each channel for each cluster).	SETUP & ISODAT
STDEV	R	Standard deviations (one for each channel for each cluster).	ISODAT
NOFLD	I	Number of fields input by user.	SETUP
FLDNAM	A	Array containing alphanumeric name for each field.	SETUP
BLOCK	I	Array containing field identifiers. Begin and end scan line, scan line increment, begin and end sample point and sample increment for each field.	SETUP

FORTRAN Name	Mode	Definition	ROUTINE WHERE INITIALIZED OR CALCULATED
BLK	I	Array containing number of scan lines and number of points on a scan line for each field.	TAPERD
SYMBLS	A	Array containing symbols to be used in the printout to identify the clusters.	SETUP
DAS	I	Unit upon which the DAS output tape is to be generated.	SETUP
FORMAT	I	Trigger indicating which format the data tape is in.	SETUP
MAP	I	Print map of clustered data every MAP iterations.	SETUP
SPTRIG	I	Trigger indicating whether or not user has input a value for SEP or desires to use default option.	SETUP
IRD	I	Number of records written on drum.	TAPERD
KPTS	I	Number of data points on last record of drum.	TAPERD FRMT1 or FRMT2
NOPTS	I	Maximum number of points that can be contained in the array IDAT. All drum records on unit 4 are of length NOPTS*NOFEAT except the last one which is of length KPTS*NOFEAT.	TAPERD

FORTRAN	Mode	Definition	ROUTINE WHERE INITIALIZED OR CALCULATED
CLD	R	Array containing the distances between cluster means.	CLDIST
N	I	Array containing count of points within each cluster.	ISODAT
NBLK	I	Array containing count of points in each cluster in each field.	PRINT
PUNCH	I	Trigger indicating whether or not statistics are to be punched.	SETUP
MAXCLS	I	Maximum number of clusters.	SETUP
ICHN	I	Trigger indicating whether chaining is to be performed.	SETUP
CHNTHS	R	Chaining threshold.	SETUP
ICHAIN	I	Array containing numbers of clusters which have been chained.	CHAIN
VARSIZ	I	Word size of the covariance matrix for one cluster. NOFEAT* (NOFEAT+1)/2	TAPERD
KUNIT	I	Unit for ERIPS interface tape	SETUP

In addition to the common block PASS the procedure element defines four parameter variables. These are used for dimensions and were defined in this manner because of the high probability that the user may need to change the dimensions within the program. Changing the parameter variable in the procedure element will change it throughout the program, if all routines are recompiled.

Definitions of the parameter variables are as follows:

FORTRAN SYMBOL	DEFINITION
MAXPOP	Dimension for number of clusters. MAXCLS cannot exceed this number. Set at 50.
MAXFET	Dimension for number of features. NOFEAT cannot exceed this number. Set at 30.
MAXFLD	Dimension for number of fields. NOFLD cannot exceed this number. Set at 100.
MAXDIM	Dimension of the array IDAT. This dimension is set as large as possible to avoid using drum whenever there is enough core storage available. This number is set at 25000. It can be decreased without any effect on the execution of the program (possibly execution time may be increased since drum will have to be accessed more frequently).

The following routines include the procedure element and must be recompiled when a change is made.

1. CHAIN
2. CLDIST
3. COVARR
4. DASTAP
5. ISODAT
6. ISOCLS
7. PCHSTA
8. PRINT
9. SETUP
10. TAPERD
11. TWRITE

5.3 SUBPROGRAM DOCUMENTATION

The following subroutines were taken from other programs and are not documented here.

1. FIND12 (NXTCHR,NUMBER,FIND) taken from LARSYS
2. FLTNUM taken from LARSYS
3. UNPACK taken from LARSYS
4. UNPKIN taken from TLU

SUBROUTINE CHAIN

IDENTIFICATION

Name/Title	- CHAIN
Author/Date	- Ruth Minter, October 1972
Organization/Installation	- LEC for CAD-MSC
Machine Identification	- UNIVAC 1108
Source Language	- FORTRAN V

PURPOSE

Subroutine CHAIN chains into one cluster all those subclusters that are close to at least one other subcluster in the group. Two subclusters are said to be close if their intercluster distance is less than some prespecified threshold value.

USAGE

- Calling Sequence

  CALL CHAIN

- Data In/Out

  Labeled Common:

Block Name	Input	Output
PASS	LNCAT	ICHAIN (ID)
	CLD	SYMBLS
	CHNTHS	
	SYMBLS	
	HEAD	

For definitions of labeled common parameters see Section 5.2.

- STORAGE

  Coding occupies 250_8 (168_{10}) locations and internai data occupies 157_i (111_{10}) locations.

METHOD

The end product of this subroutine is a converged index array (IP or ICHAIN). Subclusters that are to be chained together (and thus considered to belong to one composite cluster) will have the same index value. The index array is obtained as follows:

1. Set IP(I) = I for I = 1 to LNCAT (number of clusters).
2. Set JP(I) = IP(I) for I = 1 to LNCAT.
3. Set I = 1
4. For all clusters J, J>I, if CLD(I,J)<CHNTHS set IP(J) = IP(I) = MIN (IP(I),IP(J)).
5. If I = LNCAT go to step 6, otherwise I = I + 1 return to step 4.
6. If JP(I) ≠ IP(I) for any I, I = 1 to LNCAT, return to step 2, otherwise the process has converged.

The subroutine prints a summary of the clusters which have been chained, adjusts the symbol array for the final map printout of the data and then returns to the main program.

SUBROUTINE CLDIST

IDENTIFICATION

Name/Title	– CLDIST
Author/Date	– Ruth Minter, April 1972
Organization/Installation	– LEC for CAD-MSC
Machine Identification	– UNIVAC 1108
Source Language	– FORTRAN V

PURPOSE

Subroutine CLDIST calculates weighted Euclidian distance between cluster means.

USAGE

- Calling Sequence

  CALL CLDIST

- Data In/Out

  Labeled Common:

Block Name	Input	Output
PASS	LNCAT MEANS (AMN) STDEV NOFEAT	CLD

For definitions of labeled common parameters, see Section 5.2.

ORIGINAL PAGE IS OF POOR QUALITY

- Storage

Coding occupies 142_8 (98_{10}) locations. Internal data occupies 31_8 (25_{10}) locations.

METHOD

- Symbol Definition

Text	Code	Type	Description
i,j	I,J	I	Index for specific cluster.
k	K	I	Index for specific feature,
μ	AMN	R	Means
σ	STDEV	R	Standard deviations
CLD_{ij}	CLD	R	Distance between clusters i and j .

- Model

$$CLD_{ij} = \left[\left(\mu_{ki} - \mu_{kj}\right)^2 \Big/ \sigma_{ki}\sigma_{kj}\right]^{1/2}$$

If σ_{ki} or σ_{kj} equals zero, CLD_{ij} is arbitrarily set to 999.99 to prevent the cluster being chained to other clusters. A standard deviation of zero usually indicates bad data.

CPD202

SUBROUTINE COVARR

IDENTIFICATION

Name/Title	- COVARR
Author/Date	- Ruth Minter, October 1972
Organization/Installation	- LEC for CAD-MSC
Machine Identification	- UNIVAC 1108
Source Language	- FORTRAN V

PURPOSE

Subroutine COVARR calculates and prints the covariance matrix for each cluster. Since the covariance matrix is symmetrical only the lower triangular portion of the matrix is calculated.

USAGE

- Calling Sequence

  CALL COVARR (COVAR, C, IPLACE)

  Arguments:

Parameter Name	In/Out	Dimension	Type	Description
COVAR	Out	(LNCAT, VARSIZ)	R	Array containing covariance matrix for each of LNCAT clusters.
C	In	(NOFEAT, NOPTS)	R	Array containing input data. If the data exceeds the maximum dimensions of this array, it is stored on drum unit 4 and read in a block at a time.

Parameter Name	In/Out	Dimension	Type	Description
IPLACE	In	NOPTS	I	Array containing cluster number to which each corresponding data point belongs. This information is stored on drum unit 7 if the user has requested more data than the dimension limits.

- Data In/Out

Labeled Common:

Block Name	Input	Output
PASS	VARSIZ, LNCAT, NOPTS, IRD, KPTS NOFEAT, MEANS HEAD	

For definitions of labeled common parameters see section 5.2

- Storage

Coding occupies 445_8 (293_{10}) locations and internal data occupies 77_8 (63_{10}) locations.

METHOD

- Symbol Definition

Text	Code	Type	Description
C_{jk}	COVAR	R	The (j,k^{th}) element in the covariance matrix for a particular cluster.
n	N	I	The number of data points in a particular cluster
j,k	J,K	I	Indices corresponding to particular features

- Model

For each cluster the covariance matrix is obtained as follows:

$$C_{jk} = \frac{1}{n}\sum_{1}^{n} X_j X_k - \mu_j \mu_k \qquad j=1,\ \text{NOFEAT and } k=1,j$$

Each element in the covariance matrix is calculated as indicated above and stored in the array COVAR in consecutive locations.

Example for 4 features

$$C = \begin{pmatrix} C_{11} & & & \\ C_{12} & C_{22} & & \\ C_{13} & C_{23} & C_{33} & \\ C_{14} & C_{24} & C_{34} & C_{44} \end{pmatrix}$$

The matrix is stored as follows for cluster I.

COVAR (I,1) = C_{11}

(I,2) = C_{12}

(I,3) = C_{22}

(I,4) = C_{13}

(I,5) = C_{23}

(I,6) = C_{33}

(I,7) = C_{14}

(I,8) = C_{24}

(I,9) = C_{34}

(I,10) = C_{44}

In this example the parameter VARSIZ would be 10.

SUBROUTINE DASTAP

IDENTIFICATION

Name/Title	- DASTAP
Author/Date	- Ruth Minter, April 1972
Organization/Installation	- LEC for CAD-MSC
Machine Identification	- UNIVAC 1108
Source Language	- FORTRAN V

PURPOSE

Subroutine DASTAP generates the output tape for the PMIS DAS (Data Analysis Station) if requested by the user.

USAGE

- Calling Sequence

```
CALL DASTAP(IPLACE,IBUF,IOBUF)
```

Arguments:

Parameter Name	In/Out	Dimension	Type	Description
IPLACE	In	NOPTS	I	Array containing the cluster number to which each data point belongs. This information will be stored on drum unit 7 if the user has requested more data than the dimension limits.

Parameter Name	In/Out	Dimension	Type	Description
IBUF	In		I	Storage buffer used for one scan line of data taken from larger array IPLACE.
IOBUF	In		I	Storage buffer used for the packed scan line.

- Data In/Out

Labeled Common:

Block Name	Input	Output
PASS	IRD NOPTS BLK DAS KPTS NOFLD ICHAIN	None

- Storage

Coding occupies 234_8 (156_{10}) locations. Internal data occupies 70_8 (56_{10}) locations.

METHOD

Beginning with the first field input by the user, a buffer is filled with the cluster identification for one scan line of data. Subroutine PACK is called to pack the line in the 16 bit format expected by the PMIS DAS. The line is then written on the user's tape mounted on unit DAS. This is repeated until all the scan lines from one field have been packed and written on the output tape. An end-of-file is written after the field and the process is repeated until all fields have been written on the output tape.

The format of the DAS output tape is described in more detail in Section 3.3.2.

SUBROUTINE ISODAT

IDENTIFICATION

Name/Title	- ISODAT
Author/Date	- Ruth Minter, April 1972
Organization/Installation	- LEC for CAD-MSC
Machine Identification	- UNIVAC 1108
Source Language	- FORTRAN V

PURPOSE

Subroutine ISODAT performs the clustering algorithm described in the Technical Description (Section 2.2) of this document.

USAGE

- Calling Sequence

```
CALL ISODAT(C,IPLACE)
```

Arguments:

Parameter Name	In/Out	Dimension	Type	Description
C	In	NOFEAT x NOPTS	R	Array containing the data which is to be clustered. If the data exceeds the maximum dimensions of this array, it is stored on drum unit 4 and read in a block at a time when needed.

Parameter Name	In/Out	Dimension	Type	Description
IPLACE	Out	NOPTS	I	Array containing the number of the cluster to which the corresponding data point belongs. Subroutine ISODAT stores this information on drum unit 7 if it exceeds the dimension limits for IPLACE.

- Data In/Out

  Labeled Common:

Block Name	Input	Output
PASS	NOFEAT,ISTOP, LNCAT,NMIN,KRN, DLMIN,SEP,MEANS SPTRIG,IRD,KPTS, NOPTS,STDMAX,MAP, MAXCLS	MEANS,STDEV CLD,LNCAT,N

- Storage

  Coding occupies 1331_8 (729_{10}) locations. Internal data occupies 6456_8 (3374_{10}) locations.

METHOD

See the Technical Description of this program (Section 2.2) for a discussion of the algorithm implemented in this subroutine.

Parameter Name	In/Out	Dimension	Type	Description
IPLACE	Out	NOPTS	I	Array containing the number of the cluster to which the corresponding data point belongs. Subroutine ISODAT stores this information on drum unit 7 if it exceeds the dimension limits for IPLACE.

- Data In/Out

  Labeled Common:

Block Name	Input	Output
PASS	NOFEAT,ISTOP, LNCAT,NMIN,KRN, DLMIN,SEP,MEANS SPTRIG,IRD,KPTS, NOPTS,STDMAX,MAP, MAXCLS	MEANS,STDEV CLD,LNCAT,N

- Storage

  Coding occupies 1331_8 (729_{10}) locations. Internal data occupies 6456_8 (3374_{10}) locations.

METHOD

See the Technical Description of this program (Section 2.2) for a discussion of the algorithm implemented in this subroutine.

RESTRICTIONS

The restrictions given in section 4.1 apply to this subroutine.

SUBROUTINE PACK

IDENTIFICATION

Name/Title	- PACK
Author/Date	- Ruth Minter, October 1972
Organization/Installation	- LEC for CAD-MSC
Machine Identification	- UNIVAC 1108
Source Language	- FORTRAN V

PURPOSE

Subroutine PACK packs a string of bytes into 36 bit words. The byte length can be any number of bits less than 36.

USAGE

- Calling Sequence

CALL PACK(LNGTH,BIAS,NSAMP,INBUF,OBUF,WDOUT)

Arguments:

Parameter Name	In/Out	Dimension	Type	Description
LNGTH	In	1	I	Bit length of the bytes to be packed. Must be less than 36.
BIAS	In	1	I	Bit position to begin packing the data in the output buffer. Can be greater than 36.

Parameter Name	In/Out	Dimension	Type	Description
NSAMP	In	1	I	Number of bytes to be packed.
INBUF	In	NSAMP	I	Array containing the data to be packed. The data must be right justified in the last LNGTH bits of the 36 bit computer word.
OBUF	Out	WDOUT	I	Array containing the packed data. Trailing bits in the last word are meaningless.
WDOUT	Out	1	I	Number of words in the output array OBUF.

- Storage

  Coding occupies 236_8 (158_{10}) locations and internal data occupies 25_8 (21_{10}) locations.

METHOD

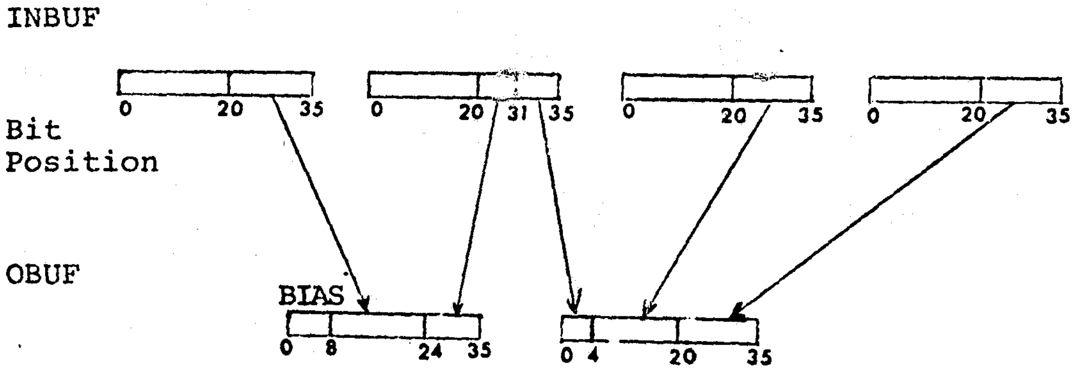
The routine uses the bit manipulation function FLD to extract the correct bits from the input buffer and insert them into the output buffer.

EXAMPLE

LNGTH = 16

BIAS = 8

NSAMP = 4



SUBROUTINE PCHSTA

IDENTIFICATION

Name/Title	- PCHSTA
Author/Date	- Ruth Minter, October 1972
Organization/Installation	- LEC for CAD-MSC
Machine Identification	- UNIVAC 1108
Source Language	- FORTRAN V

PURPOSE

Subroutine PCHSTA punches the means and covariance matrix for each cluster in two different formats. One of the formats is the same as the one used in the 1108 version of the LARSYS's STAT processor; cards punched in this format can be used in the 1108 versions of TLU and LARSYS. The other format is the same as the 360/75 ERIPS STAT processor.

USAGE

- Calling Sequence

```
CALL PCHSTA(LOCK,COVAR)
```

Arguments:

Parameter Name	In/Out	Dimension	Type	Description
LOCK	In/Out	(VARSIZ/5) *14	A	Storage area used to encode the output data in preparation for punching cards for the 360/75.

Parameter Name	In/Out	Dimension	Type	Description
COVAR	In	(LNCAT, VARSIZ)	R	Covariance matrices for all clusters.

- Data In/Out

Labeled Common:

Block Name	Input
PASS	LNCAT,VARSIZ, NOFEAT,PUNCH MEANS

For definitions of labeled common parameters see section 5.2.

- Storage

Coding occupies 456_8 (302_{10}) locations and internal data occupies 121_8 (81_{10}) locations.

METHOD

For the 1108 programs the statistics are punched with a 5E15.8 format with the keyword 'MEAN' or 'COVAR' in the first 5 columns of each card.

Since the 1108 'E' format specification is not compatible with the 360/75 it was necessary to make two alterations in the 1108 card output so that the 360/75 could read the cards. The 360/75 version of LARSYS punches the statistics with an E14.7 format with the key letters

'MN' or 'CV' in the first two columns and a sequence number in the last 8 columns of the card. The differences in the two 'E' format specifications are as follows:

1108 E14.7 ±.XXXXXXX±XX
On output the plus sign preceding the number (but not the exponent) is omitted.

360/75 E14.7 ±.XXXXXXXE±XX
On output the plus sign is omitted from both the number and the exponent.

Since the 360/75 expects the exponent to have an E preceding it, it was necessary to encode the numbers with an E14.8 format and replace the last decimal digit with an 'E'. In addition all plus signs are removed (replaced with a blank) because of a difference in key punch.

SUBROUTINE PRINT

IDENTIFICATION

Name/Title	- PRINT
Author/Date	- Ruth Minter, April 1972
Organization/Installation	- LEC for CAD-MSC
Machine Identification	- UNIVAC 1108
Source Language	- FORTRAN V

PURPOSE

Subroutine PRINT provides most of the printed output for the program. PRINT is called by subroutine ISODAT, the number of times it is called is controlled by the input parameters KRN and MAP. The main program calls PRINT for the final printout.

USAGE

- Calling Sequence

  CALL PRINT(KKT,IPLACE)

  Arguments:

Parameter Name	In/Out	Dimension	Type	Description
KKT	In	1	I	Iteration counter.

Parameter Name	In/Out	Dimension	Type	Description
IPLACE	In	NOPTS	I	Array containing the cluster number to which the corresponding data point belongs. This information will be stored on drum unit 7 if the data exceeds dimension limits.

- Data In/Out

Labeled Common:

Block Name	Input	Output
PASS	HEAD,NOFEAT, FETVEC,LNCAT MEANS,STDEV, NOFLD,FLDNAM, BLOCK,BLK, SYMBLS,IRD, KPTS,NOPTS, CLD,MAP	BLK

- Storage

Coding occupies 1035_8 (541_{10}) locations. Internal data occupies 1166_8 (630_{10}) locations.

METHOD

The following information is printed by this subroutine each time it is called.

(1) Header record

(2) Toŧal number of clusters

(3) Total number of data points

(4) Summary of total points in each cluster

(5) Means

(6) Standard deviations

(7) Distances between clusters

If the user has requested a map of the data to be printed for this iteration or if the iteration counter has been set negative then the following is printed for each field.

(a) Header record

(b) Field name

(c) Total number of points in the field

(d) Field boundaries (lines and samples)

(e) For each data point in the field, the symbol associated with the cluster to which the point belongs

(f) Summary of points per cluster in the field.

If the number of sample points on a scan line exceeds 110, only the first 110 points are printed on the line printer.

SUBROUTINE SETUP

IDENTIFICATION

Name/Title	- SETUP
Author/Date	- Ruth Minter, April 1972
Organization/Installation	- LEC for CAD-MSC
Machine Identification	- UNIVAC 1108
Source Language	- FORTRAN V

PURPOSE

Subroutine SETUP reads and analyzes all card input to the program, initializes default options and assigns drum length to units 4 and 7.

USAGE

- Calling Sequence

  CALL SETUP

- Data In/Out

  Labeled Common:

Block Name	Output
PASS	HEAD,NOFEAT,FETVEC,ISTOP LNCAT,NMIN,KRN,DLMIN,SEP MEANS,NOFLD,FLDNAM,BLOCK, SYMBLS,DAS,SPTRIG,FORMAT, MAP,PUNCH,STDMAX,MAXCLS ICHN, CHNTHS

SUBROUTINE SETUP

IDENTIFICATION

Name/Title	- SETUP
Author/Date	- Ruth Minter, April 1972
Organization/Installation	- LEC for CAD-MSC
Machine Identification	- UNIVAC 1108
Source Language	- FORTRAN V

PURPOSE

Subroutine SETUP reads and analyzes all card input to the program, initializes default options and assigns drum length to units 4 and 7.

USAGE

- Calling Sequence

CALL SETUP

- Data In/Out

Labeled Common:

Block Name	Output
PASS	HEAD,NOFEAT,FETVEC,ISTOP
	LNCAT,NMIN,KRN,DLMIN,SEP
	MEANS,NOFLD,FLDNAM,BLOCK,
	SYMBLS,DAS,SPTRIG,FORMAT,
	MAP,PUNCH,STDMAX,MAXCLS
	ICHN, CHNTHS

- Storage

  Coding occupies 1362_8 (754_{10}) locations. Internal data occupies 606_8 (390_{10}) locations.

METHOD

Card input to the program is in three formats. The first cards in the input deck (control cards) are identified by key words in columns 1-6 of the card. These key-words are listed in section 3.1.1. The subroutine reads the data card, identifies the key-word and branches to the statement number which processes that particular input. After the information located in columns 11-80 of the card is processed the program branches back to the read statement and reads the next card.

The key-word *END* signals the end of the control cards. If the control parameter LNCAT has been set greater than 1, the initial estimated cluster means are read with an 8F10.3 format. If LNCAT is less than or equal to 1, the read statement is not executed.

Immediately following the means (or the *END* card if means not input), the field identifiers are read. A field is completely defined on one card, the name is read with an A6 format from columns 1-6, the boundaries are read with an I5 format (see section 3.1.1 for a more detailed format description). Up to 'MAXFLD' (this parameter variable is set at 20) fields may be read in. The key-word $END* signals the end of all card input.

After all card input has been read, the number of data points (IPTS) user has requested is calculated from the field boundaries. This computation is necessary for assigning drum lengths.

The user's input data will be stored unpakced on drum unit 4 by subroutine TAPERD, the length of drum unit 4 must be IPTS*NOFEAT words. Subroutine ISODAT will store on drum unit 7 the cluster number to which each data point belongs, so drum unit 7 must have a word length of IPTS.

The system routines DRMAVL and DRMASG allow Univac 1108 EXEC 2 users flexibility in the assignment of scratch drum files. The system default divides the FH432 drum into 4 equal drum files of length $600{,}000_8$ words and the FH1782 drum is not used. DRMASG and DRMAVL increases the drum space available by providing access to the FH1782 drum and allows the user to specify the number and length of the files.

Both file assignments in this program (4 and 7) are made to FH432 drum if possible, since this is a higher speed drum than FH1782. However the user is allowed all of FH432 drum for the data (Unit 4) if needed, in this case Unit 7 will be assigned to the FH1782 drum.

SUBROUTINE TAPERD

IDENTIFICATION

Name/Title	- TAPERD
Author/Date	- Ruth Minter, April 1972
Organization/Installation	- LEC for CAD-MSC
Machine Identification	- UNIVAC 1108
Source Language	- FORTRAN V

PURPOSE

Subroutine TAPERD reads the users data tape and calls the necessary routines to unpack the data, select the data requested by the user and store the data unpacked on drum unit 4.

USAGE

- Calling Sequence

CALL TAPERD(IBUF)

Arguments:

Parameter Name	In/Out	Dimension	Type	Description
IBUF	Out	MAXDIM	I	Storage array passed from the main program (called IDAT in the main program). This array will at one time contain one scan line of data packed as read from

Parameter Name	In/Out	Dimension	Type	Description
				tape and the line unpacked 8 bit elements per word, the remainder of the array is used to accumulate the data user has requested in the field definitions.

- Data In/Out

Labeled Common:

Block Name	Input	Output
PASS	NOFLD,BLOCK	IRD,KPTS,BLK,NOPTS
	FETVEC,NOPTS	VARSIZ

- Storage

Coding occupies 564_8 (372_{10}) locations. Internal data occupies 241_8 (161_{10}) locations.

METHOD

Since the storage requirements vary considerably with the data tapes used, dimensions for the buffer areas needed are calculated after reading the header record from the user's input tape.

Immediately upon entry into the subroutine, the header record is read from the data tape and unpacked. The number of channels (NC) and the number of samples per line (NS) are extracted from the header record. The word length of the buffer to receive the packed data record from tape is determined as follows:

KBUF = NC*NS*8/36 + 1

The buffer to hold the unpacked scan line must be NC*NS words. The amount of storage left in the IBUF array is MAXDIM - KBUF - NC*NS. This storage is used to accumulate the data to be used in the clustering procedure as it is read and unpacked from the tape. When the buffer is filled it is written on to drum unit 4.

After all the data to be clustered has been stored on drum, the buffer areas used to read and unpack the data tape can be used to store the data to be clustered. The amount of storage needed for the covariance matrices is calculated (VARSIZ*MAXCLS) because this will be stored in the IBUF (or IDAT) array along with the data. The maximum number of points which the core array will hold is calculated as NOPTS. If this number is less than or equal to the number of data points to be clustered, then the data is read back into core from drum unit 4 and drum is never used again.

If all the data cannot be contained in core at one time, then it must be read in a block at a time. The number of blocks is calculated as IRD and the length of each block is NOPTS*NOFEAT except the last which is KPTS*NOFEAT.

SUBROUTINE TDATE

IDENTIFICATION

Name/Title	- TDATE
Author/Date	- Ruth Minter, April 1972
Programmer	- LEC Test Data Reduction
Organization/Installation	- LEC for CAD-MSC
Machine Identification	- UNIVAC 1108
Source Language	- SLEUTH

PURPOSE

This assembly language routine was designed to be called by FORTRAN and to return the current date DAY MONTH YEAR in the calling argument, which must be dimensioned by two.

USAGE

- Calling Sequence

```
CALL TDATE(DATE)
```

Arguments:

Parameter Name	In/Out	Dimension	Type	Description
DATE	Out	2	A	Array which contains the current date in the form DAY MONTH YEAR upon return to the calling routine.

5.4 Program Listing

```
ØL PDP,* COMMON.COMMON                                                                  03 DEC 73            12:22:44.477
PDP  BLI  2403 0010
THIS PROC ELEMENT PROCESSED ON 03 DEC 73 AT 12:22:44

000001         PROC ORIGIN 1      ENTRY POINT 1     COMMON*     FCOPY

000002                                                  PARAMETER MAXPOP=50,MAXFET=30

000003                                                  PARAMETER MAXDIM=25000

000004                                                  PARAMETER MAXFLD=100

000005                                                  COMMON/PASS/HEAD(42),NOFEAT,FETVEC(MAXFET),ISTOP,LNCAT,NMIN,KRN,

000006                                                 * STDMAX,DLMIN,SEP,MEANS(MAXFET,MAXPOP),STDEV(MAXFET,MAXPOP),

000007                                                 * NOFLD,FLDNAM(MAXFLD),BLOCK(MAXFLD,6),BLK(MAXFLD,2),

000008                                                 * SYMBLS(MAXPOP),DAS,FORMAT,MAP,

000009                                                 * SPTRIG,IRD,KPTS,NOPTS,CLD(MAXPOP,MAXPOP)

000010                                                 *, N(MAXPOP),NBLK(MAXFLD,MAXPOP)

000011                                                 * ,PUNCH,MAXCLS,ICHN,CHNTHS,ICHAIN(MAXPOP),VARSIZ

000012                                                 *,KUNIT

000013                                                  INTEGER VARSIZ

000014                                                  INTEGER PUNCH

000015                                                  REAL MEANS

000016                                                  INTEGER SPTRIG,BLOCK,BLK,SYMBLS,HEAD,FETVEC,FLDNAM,DAS

000017                                                  INTEGER FORMAT

000018                                                  END
   COMMON          PROCEDURE                                        29 NOV 73  13:40:47   0  01536122    14     18  (DELETED)
                                                                                          1  01536216    12     1
```

```
@ FOR,* CHAIN,CHAIN                                                                                03 DEC 73          12:22:44.968
UNIVAC 1108 FORTRAN V EXEC II LEVEL 25A  -(EXEC8 LEVEL E1201001DA)
THIS COMPILATION WAS DONE ON 03 DEC 73 AT 12:22:45

   SUBROUTINE CHAIN      ENTRY POINT 000242

   STORAGE USED: CODE(1) 000250; DATA(0) 000157; BLANK COMMON(2) 000000

    COMMON BLOCKS:

    0003    PASS    026574

   EXTERNAL REFERENCES (BLOCK, NAME)

    0004    NWDU$
    0005    NI02$
    0006    NI01$
    0007    NERR3$

   STORAGE ASSIGNMENT  (BLOCK, TYPE, RELATIVE LOCATION, NAME)

    0001    000203 100L      0000    000071 110F      0000    000103 120F      0001    000005 122G      0001    000016 127G
    0000    000125 130F      0000    000142 140F      0001    000044 141G      0001    000070 153G      0001    000112 166G
    0001    000012 20L       0001    000171 214G      0001    000022 40L       0001    000060 50L       0001    000064 60L
    0001    000103 80L       0001    000133 90L       0003 I 007305 BLK       0003 I 006155 BLOCK     0003 R 026507 CHNTHS
    0003 R 007706 CLD       0003 I 007677 DAS       0003    000116 DLMIN     0003 I 000053 FETVEC    0003 I 006011 FLDNAM
    0003 I 007700 FORMAT    0003 I 000000 HEAD      0000 I 000063 I         0003    026510 ICHAIN    0003    026506 ICHN
    0000 I 000062 IHD       0000 I 000070 IM        0000    000145 INJP$     0003 I 026510 IP        0003    007703 IRD
    0003    000111 ISTOP     0000 I 000065 J         0000 I 000000 JP        0000 I 000067 K         0000 I 000066 KNCAT
    0003    007704 KPTS      0003    000114 KRN       0003    026573 KUNIT     0003 I 000112 LNCAT     0000 I 000064 M
    0003    007701 MAP       0003    026505 MAXCLS    0003 R 000120 MEANS     0003    014612 N         0003    014674 NBLK
    0003    000113 NMIN      0003    000052 NOFEAT    0003    006010 NOFLD     0003    007705 NOPTS     0003 I 026504 PUNCH
    0003    000117 SEP       0003 I 007702 SPTRIG    0003    003054 STDEV     0003    000115 STDMAX    0003 I 007615 SYMBLS
    0003 I 026572 VARSIZ


00101          1*             SUBROUTINE CHAIN                                                  CHAI0000
00101          2*       C***********************************************************************CHAI0010
00101          3*       C*                                                                     *CHAI0020
00101          4*       C*    THIS SUBROUTINE CHAINS ALL CLUSTERS WHOSE MEANS ARE LESS THAN    *CHAI0030
00101          5*       C*    DLMIN UNITS APART.                                               *CHAI0040
00101          6*       C*          IF - DISTANCE BETWEEN CLUSTERS L AND M < DLMIN             *CHAI0050
00101          7*       C*               DISTANCE BETWEEN CLUSTERS L AND N > DLMIN             *CHAI0060
00101          8*       C*               DISTANCE BETWEEN CLUSTERS M AND N < DLMIN             *CHAI0070
00101          9*       C*          THEN-CLUSTERS L,M, AND N ARE CHAINED                       *CHAI0080
00101         10*       C*                                                                     *CHAI0090
00101         11*       C*    INPUT  CLD-CLUSTER DISTANCES                                     *CHAI0100
00101         12*       C*           DLMIN-MINIMUM DISTANCE BETWEEN CLUSTERS                   *CHAI0110
00101         13*       C*           LNCAT-NUMBER OF CLUSTERS                                  *CHAI0120
00101         14*       C*                                                                     *CHAI0130
00101         15*       C*    OUTPUT   ICHAIN-ARRAY CONTAINING NUMBERS OF CHAINED CLUSTERS     *CHAI0140
00101         16*       C*             PRINTED SUMMARY OF CLUSTERS WHICH WERE CHAINED          *CHAI0150
```

ORIGINAL PAGE IS
OF POOR QUALITY

```
00101   17*    C***********************************************************************CHAI0160
00101   18*    C*                                                                      CHAI0170
00103   19*          INCLUDE COMMON,LIST                                               CHAI0180
00104   19*          PARAMETER MAXPOP=50,MAXFET=30
00105   19*          PARAMETER MAXDIM=25000
00106   19*          PARAMETER MAXFLD=100
00107   19*          COMMON/PASS/HEAD(42),NOFEAT,FETVEC(MAXFET),ISTOP,LNCAT,NMIN,KMN,
00107   19*         * STDMAX,DLMIN,SEP,MEANS(MAXFET,MAXPOP),STDEV(MAXFET,MAXPOP),
00107   19*         * NOFLD,FLDNAM(MAXFLD),BLOCK(MAXFLD,6),BLK(MAXFLD,2),
00107   19*         * SYMBLS(MAXPOP),DAS,FORMAT,MAP,
00107   19*         * SPTRIG,IRD,KPTS,NOPTS,CLD(MAXPOP,MAXPOP)
00107   19*         *, N(MAXPOP),NBLK(MAXFLD,MAXPOP)
00107   19*         * ,PUNCH,MAXCLS,ICHN,CHNTHS,ICHAIN(MAXPOP),VARSIZ
00107   19*         *,KUNIT
00110   19*          INTEGER VARSIZ
00111   19*          INTEGER PUNCH
00112   19*          REAL MEANS
00113   19*          INTEGER SPTRIG,BLOCK,BLK,SYMBLS,HEAD,FETVEC,FLDNAM,DAS
00114   19*          INTEGER FORMAT
00115   19*          END
00116   20*          DIMENSION JP(MAXPOP),IP(MAXPOP)                                   CHAI0190
00117   21*          EQUIVALENCE (IP,ICHAIN)                                           CHAI0200
00120   22*          IHD=0                                                             CHAI0210
00121   23*          DO 10 I=1,LNCAT                                                   CHAI0220
00124   24*       10 IP(I)=I                                                           CHAI0230
00126   25*       20 DO 30 I=1,LNCAT                                                   CHAI0240
00131   26*       30 JP(I)=IP(I)                                                       CHAI0250
00133   27*          I=0                                                               CHAI0260
00134   28*       40 I=I+1                                                             CHAI0270
00135   29*          IF (I.GE.LNCAT) GO TO 60                                          CHAI0280
00137   30*          M=I+1                                                             CHAI0290
00140   31*          DO 50 J=M,LNCAT                                                   CHAI0300
00143   32*          IF(CLD(I,J).GT.CHNTHS)GO TO 50
00145   33*          IP(I)=MIN0(IP(I),IP(J))                                           CHAI0320
00146   34*          IP(J)=IP(I)                                                       CHAI0330
00147   35*       50 CONTINUE                                                          CHAI0340
00151   36*          GO TO 40                                                          CHAI0350
00152   37*       60 DO 70 I=1,LNCAT                                                   CHAI0360
00155   38*          IF (IP(I).NE.JP(I)) GO TO 20                                      CHAI0370
00157   39*       70 CONTINUE                                                          CHAI0380
00161   40*          M=1                                                               CHAI0390
00162   41*          KNCAT=LNCAT                                                       CHAI0400
00163   42*       80 K=0                                                               CHAI0410
00164   43*          IM=M + 1                                                          CHAI0420
00165   44*          DO 90 I=IM,LNCAT                                                  CHAI0430
00170   45*          IF (IP(I).NE.M) GO TO 90                                          CHAI0440
00172   46*          KNCAT=KNCAT-1                                                     CHAI0450
00173   47*          K=K+1                                                             CHAI0460
00174   48*          SYMBLS(I)=SYMBLS(M)                                               CHAI0470
00175   49*          JP(K)=I                                                           CHAI0480
00176   50*       90 CONTINUE                                                          CHAI0490
00200   51*          IF (K.EQ.0) GO TO 100                                             CHAI0500
00202   52*          IF (IHD.EQ.0) WRITE (6,140)                                       CHAI0510
00205   53*          IF (IHD.EQ.0) WRITE (6,HEAD)                                      CHAI0520
00210   54*          IHD=1                                                             CHAI0530
00211   55*          WRITE (6,110)M,(JP(I),I=1,K)                                      CHAI0540
00220   56*          WRITE (6,120)M                                                    CHAI0550
00223   57*      100 M=M+1                                                             CHAI0560
```

```
00224     58*          IF (M.LT.LNCAT) GO TO 80                                       CHAI0570
00226     59*          IF(KNCAT.EQ.LNCAT)RETURN                                       CHAI0580
00230     60*          WRITE (6,130)KNCAT                                             CHAI0590
00233     61*          RETURN                                                         CHAI0600
00234     62*      110 FORMAT(/' THE FOLLOWING CLUSTERS SHOULD BE CHAINED---',20I4)   CHAI0610
00235     63*      120 FORMAT(/' IN THE FINAL OUTPUT MAP ALL OF THE ABOVE CLUSTERS WILL BCHAI0620
00235     64*         *E REPRESENTED BY THE SYMBOL FOR CLUSTER',I4//)                 CHAI0630
00236     65*      130 FORMAT(' THE ABOVE CHAINING REDUCES THE EFFECTIVE NUMBER OF CLUSTECHAI0640
00236     66*         *RS TO ',I5)                                                    CHAI0650
00237     67*      140 FORMAT(1H1)                                                    CHAI0660
00240     68*          END                                                            CHAI0670

     END OF COMPILATION:          NO  DIAGNOSTICS.
   CHAIN          SYMBOLIC                                 29 NOV 73  13:40:49  0  01536532   14    68  (DELETED)
   CHAIN   CODE   RELOCATABLE                              29 NOV 73  13:40:49  1  01540422   24     1  (DELETED)
                                                                                0  01540452   14    22
```

ORIGINAL PAGE IS
OF POOR QUALITY

```
@ FOR,* CLDIST,CLDIST                                                                              03 DEC 73          12:22:46.8
UNIVAC 1108 FORTRAN V EXEC II LEVEL 25A  -(EXEC8 LEVEL E12010010A)
THIS COMPILATION WAS DONE ON 03 DEC 73 AT 12:22:46


    SUBROUTINE CLDIST     ENTRY POINT 000137


    STORAGE USED: CODE(1) 000153; DATA(0) 000035; BLANK COMMON(2) 000000

     COMMON BLOCKS:

     0003    PASS    026574


    EXTERNAL REFERENCES (BLOCK, NAME)

     0004    SQRT
     0005    NERR3$


    STORAGE ASSIGNMENT  (BLOCK, TYPE, RELATIVE LOCATION, NAME)

     0001    000010 121G       0001    000033 124G       0001    000047 130G       0001    000107 15L        0001    000067 5L
     0003 R  000120 AMN        0003 I  007305 BLK        0003 I  006155 BLOCK      0003    026507 CHNTHS     0003 R  007706 CLD
     0003 I  007677 DAS        0003    000116 DLMIN      0003 I  000053 FETVEC     0003 I  006011 FLDNAM     0003 I  007700 FORMAT
     0003 I  000000 HEAD       0000 I  000000 I          0003    026510 ICHAIN     0003    026506 ICHN       0000    000006 INJP$
     0003    007703 IRD        0003    000111 ISTOP      0000 I  000001 J          0000 I  000002 K          0003    007704 KRTS
     0003    000114 KRN        0003    026573 KUNIT      0003 I  000112 LNCAT      0003    007701 MAP        0003    026505 MAXCLS
     0003 R  000120 MEANS      0003    014612 N          0003    014674 NBLK       0003    000113 NMIN       0003 I  000052 NOFEAT
     0003    006010 NOFLD      0003    007705 NOPTS      0003 I  026504 PUNCH      0003    000117 SEP        0003 I  007702 SRTRIG
     0003 R  003054 STDEV      0003    000115 STDMAX     0003 I  007615 SYMBLS     0003 I  026572 VARSIZ


00100        1*      C*                                                                      CLDI0000
00100        2*      C***********************************************************************CLDI0010
00100        3*      C*                                                                     *CLDI0020
00100        4*      C*      THIS SUBROUTINE CALCULATES THE WEIGHTED DISTANCE BETWEEN       *CLDI0030
00100        5*      C*      CLUSTER MEANS                                                   CLDI0040
00100        6*      C*                                                                      CLDI0050
00100        7*      C*                                                                     *CLDI0060
00100        8*      C*      INPUT     AMN(MEANS) - MEANS OF EACH FEATURE OF EACH CLUSTER   *CLDI0070
00100        9*      C*                STDEV      - STANDARD DEVIATIONS FOR EACH FEATURE/CLUSTERCLDI0080
00100       10*      C*                LNCAT      - NUMBER OF CLUSTERS                      *CLDI0090
00100       11*      C*                NOFEAT     - NUMBER OF FEATURES (CHANNELS)           *CLDI0100
00100       12*      C*                                                                     *CLDI0110
00100       13*      C*      OUTPUT    CLD        - ARRAY CONTAINING DISTANCE BETWEEN CLUSTERS *CLDI0120
00100       14*      C*                             CLD(N,M)=DISTANCE BETWEEN CLUSTERS N AND M *CLDI0130
00100       15*      C*                                                                      CLDI0140
00100       16*      C***********************************************************************CLDI0150
00100       17*      C*                                                                      CLDI0160
00101       18*              SUBROUTINE CLDIST                                               CLDI0170
00103       19*              INCLUDE COMMON,LIST                                             CLDI0180
00104       19*              PARAMETER MAXPOP=50,MAXFET=30
00105       19*              PARAMETER MAXDIM=25000
00106       19*              PARAMETER MAXFLD=100
```

```
00107    19*        COMMON/PASS/HEAD(42),NOFEAT,FETVEC(MAXFET),ISTOP,LNCAT,NMIN,KRN,
00107    19*       * STDMAX,DLMIN,SEP,MEANS(MAXFET,MAXPOP),STDEV(MAXFET,MAXPOP),
00107    19*       * NOFLD,FLDNAM(MAXFLD),BLOCK(MAXFLD,6),BLK(MAXFLD,2),
00107    19*       * SYMBLS(MAXPOP),DAS,FORMAT,MAP,
00107    19*       * SPTRIG,IRD,KPTS,NOPTS,CLD(MAXPOP,MAXPOP)
00107    19*       *, N(MAXPOP),NBLK(MAXFLD,MAXPOP)
00107    19*       * ,PUNCH,MAXCLS,ICHN,CHNTHS,ICHAIN(MAXPOP),VARSIZ
00107    19*       *,KUNIT
00110    19*        INTEGER VARSIZ
00111    19*        INTEGER PUNCH
00112    19*        REAL MEANS
00113    19*        INTEGER SPTRIG,BLOCK,BLK,SYMBLS,HEAD,FETVEC,FLDNAM,DAS
00114    19*        INTEGER FORMAT
00115    19*        END
00116    20*        DIMENSION AMN(MAXFET,MAXPOP)                                      CLDI0190
00117    21*        EQUIVALENCE (AMN,MEANS)                                           CLDI0200
00120    22*        DO 30 I=1,LNCAT                                                   CLDI0210
00123    23*        DO 20 J=I,LNCAT                                                   CLDI0220
00126    24*        CLD(I,J)=0.0                                                      CLDI0230
00127    25*        DO 10 K=1,NOFEAT                                                  CLDI0240
00132    26*        IF(STDEV(K,I).GT.0.0  .AND.   STDEV(K,J).GT.0.0)GO TO 5
00134    27*        CLD(I,J)=999.99
00135    28*        GO TO 15
00136    29*      5 CONTINUE
00137    30*        CLD(I,J)=CLD(I,J)+(AMN(K,I)-AMN(K,J))**2/(STDEV(K,I)*STDEV(K,J))  CLDI0250
00140    31*     10 CONTINUE                                                          CLDI0260
00142    32*        CLD(I,J)=SQRT(CLD(I,J))                                           CLDI0270
00143    33*     15 CONTINUE
00144    34*        CLD(J,I)=CLD(I,J)                                                 CLDI0280
00145    35*     20 CONTINUE                                                          CLDI0290
00147    36*     30 CONTINUE                                                          CLDI0300
00151    37*        RETURN                                                            CLDI0310
00152    38*        END                                                               CLDI0320

     END OF COMPILATION:          NO  DIAGNOSTICS.
   CLDIST          SYMBOLIC                                   29 NOV 73  13:40:50  0  01541136    14    38  (DELETED)
   CLDIST  CODE    RELOCATABLE                                29 NOV 73  13:40:50  1  01542162    24     1  (DELETED)
                                                                                   0  01542212    14    12
```

```
@ FOR,* COVARR,COVARR                                                                                03 DEC 73          12:22:4
UNIVAC 1108 FORTRAN V EXEC II LEVEL 25A  -(EXEC8 LEVEL E1201001OA)
THIS COMPILATION WAS DONE ON 03 DEC 73 AT 12:22:48

   SUBROUTINE COVARR    ENTRY POINT 000423

   STORAGE USED: CODE(1) 000445; DATA(0) 000077; BLANK COMMON(2) 000000

    COMMON BLOCKS:

    0003    PASS   026574

   EXTERNAL REFERENCES (BLOCK, NAME)

    0004    NTRAN
    0005    NWDU$
    0006    NI02$
    0007    NI01$
    0010    NERR3$

   STORAGE ASSIGNMENT  (BLOCK, TYPE, RELATIVE LOCATION, NAME)

    0000     000025 100F        0000     000027 110F        0000     000030 120F        0001     000020 121G        0001     000020 124G
    0001     000120 145G        0001     000134 152G        0001     000143 155G        0001     000176 171G        0001     000211 175G
    0001     000036 20L         0001     000220 200G        0001     000260 215G        0001     000271 223G        0001     000313 233G
    0001     000347 241G        0001     000072 30L         0000     000016 90F         0003 I 007305 BLK         0003 I 006155 BLOCK
    0003     026507 CHNTHS      0003     007706 CLD         0003 I 007677 DAS         0003     000116 DLMIN       0003 I 000053 FETVEC
    0003 I 006011 FLDNAM      0003 I 007700 FORMAT      0003 I 000000 HEAD        0000 I 000001 I           0000 I 000003 ICCT
    0003     026510 ICHAIN      0003     026506 ICHN        0000 I 000010 ICLS        0000 I 000012 II          0000     000040 INJP$
    0000 I 000004 IRC         0003 I 007703 IRD         0000 I 000006 ISTAT       0003 I 000111 ISTOP       0000 I 000005 IWRDS
    0000 I 000002 J           0000 I 000014 JK          0000 I 000011 K           0000 I 000013 KINC        0000 I 000007 KK
    0003 I 007704 KPTS        0003     000114 KRN         0003     026573 KUNIT       0003 I 000112 LNCAT       0000 I 000000 LOC
    0000 I 000015 M           0003     007701 MAP         0003     026505 MAXCLS      0003 R 000120 MEANS       0003 I 014612 N
    0003     014674 NBLK        0003     000113 NMIN        0003 I 000052 NOFEAT      0003     006010 NOFLD       0003 I 007705 NOPTS
    0003 I 026504 PUNCH       0003     000117 SEP         0003 I 007702 SPTRIG      0003     003054 STDEV       0003     000115 STDMAX
    0003 I 007615 SYMBLS      0003 I 026572 VARSIZ


00101       1*           SUBROUTINE COVARR(COVAR,C,IPLACE)                              COVA0000
00101       2*      C*                                                                  COVA0010
00101       3*      C*   SUBROUTINE COVARR CALCULATES AND PRINTS THE COVARIANCE MATHIX FOR COVA0020
00101       4*      C*   EACH CLUSTER                                                   COVA0030
00101       5*      C*                                                                  COVA0040
00103       6*           INCLUDE COMMON,LIST                                            COVA0050
00104       6*           PARAMETER MAXPOP=50,MAXFET=30
00105       6*           PARAMETER MAXDIM=25000
00106       6*           PARAMETER MAXFLD=100
00107       6*           COMMON/PASS/HEAD(42),NOFEAT,FETVEC(MAXFET),ISTOP,LNCAT,NMIN,KRN,
00107       6*          * STDMAX,DLMIN,SEP,MEANS(MAXFET,MAXPOP),STDEV(MAXFET,MAXPOP),
00107       6*          * NOFLD,FLDNAM(MAXFLD),BLOCK(MAXFLD,6),BLK(MAXFLD,2),
00107       6*          * SYMBLS(MAXPOP),DAS,FORMAT,MAP,
00107       6*          * SPTRIG,IRD,KPTS,NOPTS,CLD(MAXPOP,MAXPOP)
```

```
00107      6*          *, N(MAXPOP),NBLK(MAXFLD,MAXPOP)
00107      6*          * ,PUNCH,MAXCLS,ICHN,CHNTHS,ICHAIN(MAXPOP),VARSIZ
00107      6*          *,KUNIT
00110      6*            INTEGER VARSIZ
00111      6*            INTEGER PUNCH
00112      6*            REAL MEANS
00113      6*            INTEGER SPTRIG,BLOCK,BLK,SYMBLS,HEAD,FETVEC,FLDNAM,DAS
00114      6*            INTEGER FORMAT
00115      6*            END
00116      7*            DIMENSION C(NOFEAT,NOPTS)                                   COVA0060
00117      8*            DIMENSION COVAR(LNCAT,VARSIZ),IPLACE(NOPTS)                 COVA0070
00120      9*            DO 10 I=1,VARSIZ                                            COVA0080
00123     10*            DO 10 J=1,LNCAT                                             COVA0090
00126     11*         10 COVAR(J,I)=0.0                                              COVA0100
00126     12*      C*                                                                COVA0110
00126     13*      C*    REWIND DRUM UNITS                                           COVA0120
00126     14*      C*                                                                COVA0130
00131     15*            CALL NTRAN(4,10)                                            COVA0140
00132     16*            CALL NTRAN(7,10)                                            COVA0150
00133     17*            ICCT=NOPTS                                                  COVA0160
00134     18*            IRC=IRD                                                     COVA0170
00135     19*         20 IF(IRC.LE.1)ICCT=KPTS                                       COVA0180
00137     20*            IF (IRD.EQ.0) GO TO 30                                      COVA0190
00141     21*            IWRDS=ICCT*NOFEAT                                           COVA0200
00142     22*            CALL NTRAN(4,2,IWRDS,C,ISTAT,22)                            COVA0210
00143     23*            CALL NTRAN(7,2,ICCT,IPLACE,ISTAT,22)                        COVA0220
00143     24*      C*                                                                COVA0230
00143     25*      C*    SINCE THE COVARIANCE MATRIX IS SYMMETRICAL ONLY THE LOWER   COVA0240
00143     26*      C*    TRIANGULAR PORTION OF THE MATRIX IS CALCULATED.             COVA0250
00143     27*      C*                                                                COVA0260
00144     28*         30 DO 40 I=1,ICCT                                              COVA0270
00147     29*            KK=0                                                        COVA0280
00150     30*            ICLS=IPLACE(I)                                              COVA0290
00151     31*            DO 40 J=1,NOFEAT                                            COVA0300
00154     32*            DO 40 K=1,J                                                 COVA0310
00157     33*            KK=KK+1                                                     COVA0320
00160     34*            COVAR(ICLS,KK)=COVAR(ICLS,KK)+C(J,I)*C(K,I)                 COVA0330
00161     35*         40 CONTINUE                                                    COVA0340
00165     36*            IRC=IRC-1                                                   COVA0350
00166     37*            IF (IRC.GT.0) GO TO 20                                      COVA0360
00170     38*            DO 50 I=1,LNCAT                                             COVA0370
00173     39*            KK=0                                                        COVA0380
00174     40*            DO 50 J=1,NOFEAT                                            COVA0390
00177     41*            DO 50 K=1,J                                                 COVA0400
00202     42*            KK=KK+1                                                     COVA0410
00203     43*            COVAR(I,KK)=COVAR(I,KK)/N(I)-MEANS(K,I)*MEANS(J,I)          COVA0420
00204     44*         50 CONTINUE                                                    COVA0430
00210     45*            WRITE (6,120)                                               COVA0440
00212     46*            WRITE (6,HEAD)                                              COVA0450
00214     47*            DO 80 I=1,LNCAT                                             COVA0460
00217     48*            WRITE (6,90)I                                               COVA0470
00222     49*            DO 70 LOC=1,NOFEAT,12                                       COVA0480
00225     50*            ISTOP=LOC+11                                                COVA0490
00226     51*            IF(ISTOP.GT.NOFEAT)ISTOP=NOFEAT                             COVA0500
00230     52*            II=1                                                        COVA0510
00231     53*            KINC=1                                                      COVA0520
00232     54*            DO 60 J=LOC,NOFEAT                                          COVA0530
00235     55*            K=J*(J+1)/2-II+1                                            COVA0540
```

ORIGINAL PAGE IS
OF POOR QUALITY

```
00236      56*            JK=K+KINC-1                                                COVA0550
00237      57*            WRITE (6,100)(COVAR(I,M),M=K,JK)                           COVA0560
00245      58*            II=II+1                                                    COVA0570
00246      59*         60 IF(KINC.LT.ISTOP)KINC=KINC+1                               COVA0580
00251      60*            WRITE (6,110)                                              COVA0590
00253      61*         70 CONTINUE                                                   COVA0600
00255      62*         80 CONTINUE                                                   COVA0610
00257      63*            RETURN                                                     COVA0620
00260      64*         90 FORMAT(//' COVARIANCE MATRIX FOR CLUSTER',I4/)             COVA0630
00261      65*        100 FORMAT(/6X,12F9.2)                                         COVA0640
00262      66*        110 FORMAT(///)                                                COVA0650
00263      67*        120 FORMAT(1H1)                                                COVA0660
00264      68*            END                                                        COVA0670

      END OF COMPILATION:          NO  DIAGNOSTICS.
   COVARR             SYMBOLIC                             29 NOV 73  13:40:52  0  01542462   14   68  (DELETED)
   COVARR   CODE      RELOCATABLE                          29 NOV 73  13:40:52  1  01544352   24    1  (DELETED)
                                                                                0  01544402   14   33
```

```
@ FOR , DASTAP,DASTAP                                                                              03 DEC 73            12:22:50.625
UNIVAC 1108 FORTRAN V EXEC II LEVEL 25A  -(EXEC8 LEVEL E12010010A)
THIS COMPILATION WAS DONE ON 03 DEC 73 AT 12:22:50

   SUBROUTINE DASTAP     ENTRY POINT 000241

   STORAGE USED: CODE(1) 000263; DATA(C) 000107; BLANK COMMON(2) 000000

    COMMON BLOCKS:

    0003    PASS    026574

   EXTERNAL REFERENCES (BLOCK, NAME)

    0004    NTRAN
    0005    PACK
    0006    NWDUS
    0007    NI02$
    0010    NERR3$

   STORAGE ASSIGNMENT  (BLOCK, TYPE, RELATIVE LOCATION, NAME)

    0001    000041 133G       0001    000050 140G       0001    000056 143G       0001    000035 20L        0001    000120 30L
    0001    000173 35L        0001    000205 40L        0000    000013 60F        0000    000033 70F        0000    000043 80F
    0003 I  007305 BLK        0003 I  006155 BLOCK      0003    026507 CHNTHS     0003    007706 CLD        0003 I  007677 DAS
    0003    000116 DLMIN      0003 I  000053 FETVEC     0003 I  006011 FLDNAM     0003 I  007700 FORMAT     0003 I  000000 HEAD
    0000 I  000007 I          0003 I  026510 ICHAIN     0003    026506 ICHN       0000 I  000004 IFLD       0000    000067 INJP$
    0000 I  000002 IPTS       0003 I  007703 IRD        0000 I  000001 IREC       0000 I  000003 ISTAT      0003    000111 ISTOP
    0000 I  000012 IWDCNT     0000 I  000010 J          0000 I  000000 K          0000 I  000011 KP         0003 I  007704 KPTS
    0003    000114 KRN        0003    026573 KUNIT      0000 I  000005 LINES      0003    000112 LNCAT      0000 I  000006 LPTS
    0003    007701 MAP        0003    026505 MAXCLS     0003 R  000120 MEANS      0003    014612 N          0003    014674 NBLK
    0003    000113 NMIN       0003    000052 NOFEAT     0003 I  006010 NOFLD      0003 I  007705 NOPTS      0003 I  026504 PUNCH
    0003    000117 SEP        0003 I  007702 SPTRIG     0003    003054 STDEV      0003    000115 STDMAX     0003 I  007615 SYMBLS
    0003 I  026572 VARSIZ


 00100         1*      C*                                                                        DAST0000
 00100         2*      C*        THIS SUBROUTINE GENERATES A TAPE FOR DISPLAY ON THE PMIS DAS.   DAST0010
 00100         3*      C*        EACH FIELD IS ON A SEPARATE FILE OF THE TAPE.                   DAST0020
 00100         4*      C*        A RECORD CONSISTS OF THE CLUSTER NUMBERS  TO WHICH EACH REQUESTED DAST0030
 00100         5*      C*        POINT ON A PARTICULAR LINE BELONGS.  THE VALUES ARE PACKED IN   DAST0040
 00100         6*      C*        SIXTEEN BIT BYTES.                                              DAST0050
 00100         7*      C*                                                                        DAST0060
 00100         8*      C*                                                                        DAST0070
 00101         9*                SUBROUTINE DASTAP(IPLACE,IBUF,IDBUF)                            DAST0080
 00103        10*                INCLUDE COMMON,LIST                                             DAST0090
 00104        10*                PARAMETER MAXPOP=50,MAXFET=30
 00105        10*                PARAMETER MAXDIM=25000
 00106        10*                PARAMETER MAXFLD=100
 00107        10*                COMMON/PASS/HEAD(42),NOFEAT,FETVEC(MAXFET),ISTOP,LNCAT,NMIN,KRN,
 00107        10*              * STDMAX,DLMIN,SEP,MEANS(MAXFET,MAXPOP),STDEV(MAXFET,MAXPOP),
 00107        10*              * NOFLD,FLDNAM(MAXFLD),BLOCK(MAXFLD,6),BLK(MAXFLD,2),
00107     10*          * SYMBLS(MAXPOP),DAS,FORMAT,MAP,
00107     10*          * SPTRIG,IRD,KPTS,NOPTS,CLD(MAXPOP,MAXPOP)
00107     10*          *, N(MAXPOP),NBLK(MAXFLD,MAXPOP)
00107     10*          * ,PUNCH,MAXCLS,ICHN,CHNTHS,ICHAIN(MAXPOP),VARSIZ
00107     10*          *,KUNIT
00110     10*            INTEGER VARSIZ
00111     10*            INTEGER PUNCH
00112     10*            REAL MEANS
00113     10*            INTEGER SPTRIG,BLOCK,BLK,SYMBLS,HEAD,FETVEC,FLDNAM,DAS
00114     10*            INTEGER FORMAT
00115     10*            END
00116     11*            DIMENSION IBUF(1),IOBUF(1)                                 DASTO100
00117     12*            DIMENSION IPLACE(NOPTS)                                    DASTO110
00120     13*            CALL NTRAN(7,10)                                           DASTO120
00121     14*            K=0                                                        DASTO130
00122     15*            IREC=IRD                                                   DASTO140
00123     16*            IPTS=NOPTS                                                 DASTO150
00124     17*         10 IF(IREC.LE.1)IPTS=KPTS                                     DASTO160
00126     18*            IF (IRD.EQ.0) GO TO 20                                     DASTO170
00130     19*            CALL NTRAN(7,2,IPTS,IPLACE,ISTAT,22)                       DASTO180
00131     20*         20 CONTINUE                                                   DASTO190
00132     21*            DO 50 IFLD=1,NOFLD                                         DASTO200
00135     22*            LINES=BLK(IFLD,1)                                          DASTO210
00136     23*            LPTS=BLK(IFLD,2)                                           DASTO220
00137     24*            DO 40 I=1,LINES                                            DASTO230
00142     25*            DO 30 J=1,LPTS                                             DASTO240
00145     26*            K=K+1                                                      DASTO250
00146     27*            KP=IPLACE(K)                                               DASTO260
00147     28*            IBUF(J)=ICHAIN(KP)                                         DASTO270
00150     29*            IF (K.LT.IPTS) GO TO 30                                    DASTO280
00152     30*            IREC=IREC-1                                                DASTO290
00153     31*            IF (IREC.EQ.0) GO TO 30                                    DASTO300
00155     32*            IF(IREC.EQ.1)IPTS=KPTS                                     DASTO310
00157     33*            CALL NTRAN(7,2,IPTS,IPLACE,ISTAT,22)                       DASTO320
00160     34*            K=0                                                        DASTO330
00161     35*         30 CONTINUE                                                   DASTO340
00163     36*            CALL PACK(16,0,LPTS,IBUF,IOBUF,IWDCNT)                     DASTO350
00164     37*            CALL NTRAN(DAS,1,IWDCNT,IOBUF,ISTAT,22)                    DASTO360
00165     38*            IF(ISTAT.GT.0)GO TO 40
00167     39*            IF(ISTAT.NE.-2)GO TO 35
00171     40*            WRITE(6,80)DAS,I
00175     41*            CALL NTRAN(DAS,22,9,22,11)
00176     42*            RETURN
00177     43*         35 WRITE(6,60)DAS,ISTAT
00203     44*            RETURN                                                     DASTO390
00204     45*         40 CONTINUE                                                   DASTO400
00206     46*            CALL NTRAN(DAS,9)                                          DASTO410
00207     47*         50 CONTINUE                                                   DASTO420
00211     48*            WRITE (6,70)NOFLD                                          DASTO430
00214     49*            RETURN                                                     DASTO450
00215     50*         60 FORMAT(/' WRITE ON UNIT',I3,' TERMINATED ABNORMALLY'/' DAS TAPE NODASTO460
00215     51*           *T CREATED',5X,' ISTAT=',I5)
00216     52*         70 FORMAT(/1X,I6,' FILES WRITTEN ON DAS OUTPUT TAPE')         DASTO480
00217     53*         80 FORMAT(' END-OF-TAPE ON UNIT',I4,' LAST LINE WRITTEN',I5/)
00220     54*            END                                                        DASTO490

      END OF COMPILATION:        NO  DIAGNOSTICS.
```

```
@FOR,* ISOCLS,ISOCLS                                                                                    03 DEC 73                12:22:56. 85
UNIVAC 1108 FORTRAN V EXEC II LEVEL 25A  -(EXEC8 LEVEL E12010010A)
THIS COMPILATION WAS DONE ON 03 DEC 73 AT 12:22:56

   MAIN PROGRAM

   STORAGE USED: CODE(1) 000207; DATA(0) 060672; BLANK COMMON(2) 000000

    COMMON BLOCKS:

    0003    PASS    026574

   EXTERNAL REFERENCES (BLOCK, NAME)

    0004    TDATE
    0005    SETUP
    0006    TAPERD
    0007    ISODAT
    0010    COVARR
    0011    TWRITE
    0012    PCHSTA
    0013    CHAIN
    0014    PRINT
    0015    DASTAP
    0016    NSTOP$
    0017    NERR3$

   STORAGE ASSIGNMENT  (BLOCK, TYPE, RELATIVE LOCATION, NAME)

    0001     000053 1OL         0001     000120 151G        0001     000165 167G        0001     000152 30L         0001     000003 SL
    0003 I 007305 BLK         0003 I 006155 BLOCK       0003 R 026507 CHNTHS      0003 R 007706 CLD         0003 I 007677 DAS
    0003 R 000016 DATE        0003 R 000116 DLMIN       0003 I 000053 FETVEC      0003 I 006011 FLONAM      0003 I 007700 FORMAT
    0003 I 000000 HEAD        0003 R 000002 HED1        0003 R 000023 HED2        0000 I 060653 I           0003 I 026510 ICHAIN
    0003 I 026506 ICHN        0000 I 000000 IDAT        0000     060661 INJP$       0003 I 007703 IRD         0003 I 000111 ISTOP
    0000 I 060652 KDIM        0000 I 060650 KPLCE       0003 I 007704 KPTS        0003 I 000114 KRN         0003 I 026573 KUNIT
    0000 I 060651 KVAR        0000 I 060654 LDIM        0003 I 000112 LNCAT       0003 I 007701 MAP         0003 I 026505 MAXCLS
    0003 R 000120 MEANS       0003 I 014612 N           0003 I 014674 NBLK        0003 I 000113 NMIN        0003 I 000052 NOFEAT
    0003 I 006010 NOFLD       0003 I 007705 NOPTS       0003 I 026504 PUNCH       0003 R 000117 SEP         0003 I 007702 SPTRIG
    0003 R 003054 STDEV       0003 R 000115 STDMAX      0003 I 007615 SYMBLS      0003 I 026572 VARSIZ


 00100        1*      C*                                                                       ISOC0000
 00100        2*      C************************************************************************ISOC0010
 00100        3*      C*                                                                       ISOC0020
 00100        4*      C*     THIS PROGRAM PERFORMS A MODIFIED VERSION OF THE CLUSTERING        ISOC0030
 00100        5*      C*     ALGORITM (ISODATA) ORIGINALLY DEVELOPED BY BALL AND HALL OF       ISOC0040
 00100        6*      C*     STANFORD RESEARCH INSTITUTE.  THE ALGORITHM HAS BEEN MODIFIED     ISOC0050
 00100        7*      C*     ON THE RECOMMENDATIONS OF ED KAN (LEC).                           ISOC0060
 00100        8*      C*                                                                       ISOC0070
 00100        9*      C*     THE PROGRAM EXPECTS MULTISPECTRAL SCANNER DATA                    ISOC0080
 00100       10*      C*     IN EITHER THE LARSYS 22 OR THE UNIVERSAL                          ISOC0090
 00100       11*      C*     FORMAT.  THE DATA TAPE SHOULD BE ASSIGNED TO FORTRAN UNIT 3.      ISOC0100
 00100       12*      C*     DRUM UNITS 4 AND 7 ARE USED INTERNALLY BY THE PROGRAM IF THE      ISOC0110
 00100       13*      C*     AMOUNT OF DATA REQUESTED EXCEEDS THE FIXED DIMENSIONS.            ISOC0120
```

```
00100     14*     C*                                                                            ISOC0130
00100     15*     C*****************************************************************************ISOC0140
00100     16*     C*                                                                            ISOC0150
00101     17*           INCLUDE COMMON,LIST                                                     ISOC0160
00103     17*           PARAMETER MAXPOP=50,MAXFET=30
00104     17*           PARAMETER MAXDIM=25000
00105     17*           PARAMETER MAXFLD=100
00106     17*           COMMON/PASS/HEAD(42),NOFEAT,FETVEC(MAXFET),ISTOP,LNCAT,NMIN,KRN,
00106     17*          * STDMAX,DLMIN,SEP,M=ANS(MAXFET,MAXPOP),STDEV(MAXFET,MAXPOP),
00106     17*          * NOFLD,FLDNAM(MAXFLD),BLOCK(MAXFLD,6),BLK(MAXFLD,2),
00106     17*          * SYMBLS(MAXPOP),DAS,FORMAT,MAP,
00106     17*          * SPTRIG,IRD,KPTS,NOPTS,CLD(MAXPOP,MAXPOP)
00106     17*          *, N(MAXFOP),NBLK(MAXFLD,MAXPOP)
00106     17*          * ,PUNCH,MAXCLS,ICHN,CHNTHS,ICHAIN(MAXPOP),VARSIZ
00106     17*          *,KUNIT
00107     17*           INTEGER VARSIZ
00110     17*           INTEGER PUNCH
00111     17*           REAL MEANS
00112     17*           INTEGER SPT=IG,9LOCK,BLK,SYMBLS,HEAD,FETVEC,FLDNAM,DAS
00113     17*           INTEGER FOR-AT
00114     17*           END
00115     18*           DIMENSION HED1(10),HED2(10),DATE(2)                                     ISOC0170
00116     19*           DIMENSION IDAT(MAXDIM)                                                  ISOC0180
00117     20*           DATA HEAD /'(1H ,T25,60H                                                ISOC0190
00117     21*          1                   ,T100,20H                          /1H ,T25,60H      ISOC020.
00117     22*          2                                                          /1H0,T10,60HISOC0210
00117     23*          3                                                                 //)'/  ISOC0220
00121     24*           EQUIVALENCE (HED1(1),HEAD(3)),  (DATE(1),HEAD(15)),                     ISOC0230
00121     25*          *             (HED2(1),HEAD(20))                                         ISOC0240
00122     26*           DATA  HED1  /'                 MANNED SPACECRAFT CENTER                 ISOC0250
00122     27*          *         '/                                                             ISOC0260
00124     28*           DATA  HED2  /'                       HOUSTON, TEXAS                     ISOC0270
00124     29*          *         '/                                                             ISOC0280
00124     30*     C*                                                                            ISOC0290
00124     31*     C*    SET THE DATE FROM TDATE                                                 ISOC0300
00124     32*     C*                                                                            ISOC0310
00126     33*           CALL TDATE(DATE)                                                        ISOC0320
00126     34*     C*                                                                            ISOC0330
00126     35*     C*    CALL SETUP TO READ CARD INPUT AND INITIALIZE DEFAULT VALUES             ISOC0340
00126     36*     C*                                                                            ISOC0350
00127     37*         5 CALL SETUP
00127     38*     C*                                                                            ISOC0370
00127     39*     C*    CALL TAPERD TO READ DATA FROM TAPE AND STORE ON DRUM                    ISOC0380
00127     40*     C*                                                                            ISOC0390
00130     41*           CALL TAPERD(IDAT)                                                       ISOC0400
00130     42*     C*                                                                            ISOC0410
00130     43*     C*    CALL ISODAT TO PERFORM CLUSTERING                                       ISOC0420
00130     44*     C*                                                                            ISOC0430
00131     45*           KPLCE = NOPTS*NOFEAT + 1                                                ISOC0440
00132     46*           CALL ISODAT(IDAT(1),IDAT(KPLCE))                                        ISOC0450
00132     47*     C*                                                                            ISOC0460
00132     48*     C*    CALCULATE COVARIANCE MATPIX FOR EACH CLUSTER                            ISOC0470
00132     49*     C*                                                                            ISOC0480
00133     50*           KVAR=KPLCE+NOPTS                                                        ISOC0490
00134     51*           CALL COVARR(IDAT(KVAR),IDAT(1),IDAT(KPLCE))                             ISOC0500
00134     52*     C
00134     53*     C     CREATE ERIPS INPUT TAPE
00134     54*     C
```

```
00135     55*          IF(KUNIT .LE. 0)  GO TO 10
00137     56*          CALL TWRITE(IDAT(KVAR))
00137     57*     C*                                                                 ISOC0510
00137     58*     C*   PUNCH STATISTICS IF REQUESTED                                 ISOC0520
00137     59*     C*                                                                 ISOC0530
00140     60*       10 IF(PUNCH.GT.0)CALL PCHSTA(IDAT(1),IDAT(KVAR))                 ISOC0540
00140     61*     C*                                                                 ISOC0550
00140     62*     C*   CHAIN CLUSTERS WHOSE DISTANCES ARE LESS THAN DLMIN            ISOC0560
00140     63*     C*                                                                 ISOC0570
00142     64*          IF(ICHN.GT.0)CALL  CHAIN
00142     65*     C*                                                                 ISOC0590
00142     66*     C*   PRINT FINAL RESULTS                                           ISOC0600
00142     67*     C*                                                                 ISOC0610
00144     68*          CALL PRINT(-1,IDAT(KPLCE))                                    ISOC0620
00144     69*     C*                                                                 ISOC0630
00144     70*     C*   CREATE DAS OUTPUT TAPE IF DESIRED                             ISOC0640
00144     71*     C*                                                                 ISOC0650
00145     72*          IF (DAS.LE.0) GO TO 30                                        ISOC0660
00147     73*          KDIM=NOPTS+1                                                  ISOC0670
00150     74*          DO 20 I=1,NOFLD                                               ISOC0680
00153     75*       20 LDIM=MAXO(LDIM,BLK(I,2))                                      ISOC0690
00155     76*          LDIM=LDIM +KDIM + 1                                           ISOC0700
00156     77*          IF(IRD.EQ.0)CALL TRNSFR                                       ISOC0710
00160     78*          CALL DASTAP(IDAT(1),IDAT(KDIM),IDAT(LDIM))                    ISOC0720
00160     79*     C*                                                                 ISOC0730
00161     80*       30 CONTINUE                                                      ISOC0740
00162     81*          GO TO 5
00162     82*     C*                                                                 ISOC0760
00162     83*     C*   TRANSFER THE DATA IN CORE FROM IDAT(KPLCE) TO IDAT(1)         ISOC0770
00162     84*     C*                                                                 ISOC0780
00163     85*          SUBROUTINE TRNSFR                                             ISOC0790
00166     86*          DO 20 I=1,KPTS                                                ISOC0800
00171     87*       20 IDAT(I)=IDAT(KPLCE+I-1)                                       ISOC0810
00173     88*          RETURN                                                        ISOC0820
00174     89*          END                                                           ISOC0830


    END OF COMPILATION:         NO  DIAGNOSTICS.
   ISOCLS          SYMBOLIC                          29 NOV 73  13;40;58   0  01556624   14   89  (DELETED)
   ISOCLS  CODE    RELOCATABLE                       29 NOV 73  13;40;58   1  01561162   36    1  (DELETED)
                                                                           0  01561226   14   22
```

ORIGINAL PAGE IS
OF POOR QUALITY

```
@ FOR,* ISODAT,ISODAT                                                              03 DEC 73          12:22:57.494
UNIVAC 1108 FORTRAN V EXEC II LEVEL 25A  -(EXEC8 LEVEL E1201001OA)
THIS COMPILATION WAS DONE ON 03 DEC 73 AT 12:22:57

  SUBROUTINE ISODAT     ENTRY POINT 001240

  STORAGE USED: CODE(1) 001371; DATA(0) 006460; BLANK COMMON(2) 000000

   COMMON BLOCKS:

   0003    PASS    026574

  EXTERNAL REFERENCES (BLOCK, NAME)

   0004    RESET
   0005    NTRAN
   0006    CLOCK
   0007    CLDIST
   0010    PRINT
   0011    NWDU$
   0012    NIO2$
   0013    SQRT
   0014    NERR3$

  STORAGE ASSIGNMENT  (BLOCK, TYPE, RELATIVE LOCATION, NAME)

  0001     000035 10L        0001     000305 110L       0000     006237 120F       0001     000017 127G       0001     000021 133G
  0001     000376 150L       0001     000406 170L       0001     000134 171G       0001     000431 190L       0001     000071 20L
  0001     000167 200G       0001     000446 200L       0001     000174 204G       0000     006243 210F       0001     000452 220L
  0001     000252 224G       0001     000457 230L       0001     000333 250G       0001     000521 250L       0001     000346 254G
  0001     000600 280L       0000     006230 30F        0001     000417 300G       0001     000617 310L       0001     000623 320L
  0001     000471 334G       0000     006263 340F       0001     000510 341G       0001     000647 350L       0001     000652 360L
  0001     000701 380L       0000     006301 390F       0001     000127 40L        0001     000733 400L       0001     000737 410L
  0001     000670 416G       0001     000755 430L       0001     001034 440L       0001     000747 450G       0001     001016 467G
  0001     001111 470L       0000     006316 490F       0001     001127 500L       0001     001057 506G       0001     001172 520L
  0001     001115 526G       0001     001174 530L       0001     001142 537G       0000     006333 540F       0001     001147 546G
  0001     001210 550L       0000     006351 560F       0001     001310 607G       0001     001311 612G       0001     000211 70L
  0001     000216 80L        0000 R 003100 AMN        0000 R 000000 AVP        0003 I 007305 BLK        0003 I 006155 BLOCK
  0003 R 026507 CHNTHS     0003 R 007706 CLD        0003 I 007677 DAS        0000 R 006214 DIST       0003 R 000116 DLMIN
  0000 R 006225 DMIN       0000 R 003016 DN         0003 I 000053 FETVEC     0003 I 006011 FLDNAM     0003 I 007700 FORMAT
  0003 I 000000 HEAD       0000 I 006202 I          0000 I 006206 ICCT       0003 I 026510 ICHAIN     0003 I 026506 ICHN
  0000 I 006224 INC        0003 I 000112 INCAT      0000   006402 INJP$      0000   006406 INJP$      0000 I 006120 IP
  0000 I 006207 IRC        0003 I 007703 IRD        0000 I 002734 ISGMA      0000 I 006220 ISPLT      0000 I 006211 ISTAT
  0003 I 000111 ISTOP      0000 I 006204 ITER       0000 I 006210 IWRDS      0000 I 006203 J          0000 I 006215 K
  0003 I 000052 KDIM       0000 I 006212 KK         0000 I 006205 KKT        0003 I 007704 KPTS       0003 I 000114 KRM
  0003 I 026573 KUNIT      0000 I 006223 KX         0000 I 006227 L          0003 I 000112 LNCAT      0000 I 006226 M
  0003 I 007701 MAP        0003 I 026505 MAXCLS     0003 R 000120 MEANS      0003 I 014612 N          0003 I 014674 NBLK
  0000 I 006222 NCAT       0003 I 000113 NMIN       0003 I 000052 NOFEAT     0003 I 006010 NOFLD      0003 I 007705 NOPTS
  0003 I 026504 PUNCH      0000 R 006216 RNO        0000 R 006213 SDIST      0003 R 000117 SEP        0000 R 006034 SGMA
  0003   000115 SGMIN      0003 I 007702 SPTRIG     0003 R 003054 STDEV      0003 R 000115 STDMAX     0003 I 007615 SYMBLS
  0000 R 006221 TEST       0000 R 006217 TIME       0000 I 006116 TRIG1      0000 I 006117 TRIG2      0003 I 026572 VARSIZ
```

```
      SUBROUTINE ISODAT(C,IPLACE)                                       ISO00000
      INCLUDE COMMON,LIST                                               ISO00010
      PARAMETER MAXPOP=50,MAXFET=30
      PARAMETER MAXDIM=25000
      PARAMETER MAXFLD=100
      COMMON/PASS/HEAD(42),NOFEAT,FETVEC(MAXFET),ISTOP,LNCAT,NMIN,KRN,
     • STDMAX,DLMIN,SEP,MEANS(MAXFET,MAXPOP),STDEV(MAXFET,MAXPOP),
     • NOFLD,FLDNAM(MAXFLD),BLOCK(MAXFLD,6),BLK(MAXFLD,2),
     • SYMBLS(MAXPOP),DAS,FORMAT,MAP,
     • SPTRIG,IRD,KPTS,NOPTS,CLD(MAXPOP,MAXPOP)
     •, N(MAXPOP),NBLK(MAXFLD,MAXPOP)
     • ,PUNCH,MAXCLS,ICHN,CHNTHS,ICHAIN(MAXPOP),VARSIZ
     •,KUNIT
      INTEGER VARSIZ
      INTEGER PUNCH
      REAL MEANS
      INTEGER SPTRIG,BLOCK,BLK,SYMBLS,HEAD,FETVEC,FLDNAM,DAS
      INTEGER FORMAT
      END
      EQUIVALENCE (SGMIN,STDMAX)                                        ISO00020
      DIMENSION AVP(MAXFET,MAXPOP)                                      ISO00030
      DIMENSION ISGMA(MAXPOP),DN(MAXPOP)                                ISO00040
      DIMENSION C(NOFEAT,NOPTS),IPLACE(NOPTS),AMN(MAXFET,MAXPOP),       ISO00050
     • SGMA(MAXPOP)                                                     ISO00060
      EQUIVALENCE (KDIM,NOFEAT),(LNCAT,INCAT)                           ISO00070
      INTEGER TRIG1,TRIG2                                               ISO00080
      DIMENSION IP(MAXPOP)                                              ISO00090
      TRIG2=0                                                           ISO00100
      DO 5 I=1,MAXCLS
      N(I)=0
      DO 5 J=1,NOFEAT
      AMN(J,I)=0.0
    5 AVP(J,I)=0.0
      ASSIGN 230 TO ITER                                                ISO00110
      KKT=1                                                             ISO00120
C•                                                                      ISO00130
C•    ASSIGN DATA TO CLUSTERS                                           ISO00140
C•                                                                      ISO00150
   10 CONTINUE                                                          ISO00160
      IF (LNCAT.LE.1.AND.KKT.GT.1) GO TO 530                            ISO00170
      CALL RESET                                                        ISO00180
      CALL NTRAN(4,10)                                                  ISO00190
      CALL NTRAN(7,10)                                                  ISO00200
      ICCT=NOPTS                                                        ISO00210
      IRC=IRD                                                           ISO00220
   20 IF(IRC.LE.1)ICCT=KPTS                                             ISO00230
      IF (IRD.EQ.0) GO TO 40                                            ISO00240
      IWRDS=NOFEAT*ICCT                                                 ISO00250
      CALL NTRAN(4,2,IWRDS,C,ISTAT,22)                                  ISO00260
      IF (ISTAT.GE.0) GO TO 40                                          ISO00270
      WRITE (6,30)ISTAT                                                 ISO00280
   30 FORMAT(' ERROR READING DRUM---ISTAT=',I4)                         ISO00290
   40 CONTINUE                                                          ISO00300
      DO 100 I=1,ICCT                                                   ISO00310
      KK=1                                                              ISO00320
      IF (LNCAT.LT.2) GO TO 80                                          ISO00330
      SDIST=10.0E+20                                                    ISO00340
```

ORIGINAL PAGE IS
OF POOR QUALITY

```
00177            DO 70 J=1,LNCAT                                                IS000360
00202            DIST=0.0                                                       IS000360
00203            DO 50 K=1,NOFEAT                                               IS000370
00206         50 DIST=DIST + ABS(MEANS(K,J)-C(K,I))                             IS000380
00210            IF (DIST-SDIST) 60,70,70                                       IS000390
00213         60 KK=J                                                           IS000400
00214            SDIST=DIST                                                     IS000410
00215         70 CONTINUE                                                       IS000420
00217         80 CONTINUE                                                       IS000430
00220            N(KK)=N(KK)+1                                                  IS000440
00221            IPLACE(I)=KK                                                   IS000450
00222            RND=(FLOAT(N(KK))-1.)/FLOAT(N(KK))                             IS000460
00223            DO 90 K=1,NOFEAT                                               IS000470
00226            AMN(K,KK) = RND*AMN(K,KK) +        C(K,I)/FLOAT(N(KK))         IS000480
00227            AVP(K,KK) = RND*AVP(K,KK) +        C(K,I) **2/FLOAT(N(KK))     IS000490
00230         90 CONTINUE                                                       IS000500
00232        100 CONTINUE                                                       IS000510
00234            IF (IRD.EQ.0) GO TO 110                                        IS000520
00236            CALL NTRAN(7,1,ICCT,IPLACE,ISTAT,22)                           IS000530
00237        110 IRC=IRC-1                                                      IS000540
00240            IF (IRC.GT.0) GO TO 20                                         IS000550
00242            CALL CLOCK(TIME)                                               IS000560
00243            WRITE (6,120)TIME                                              IS000570
00246        120 FORMAT(' TIME 1=',F7.3)                                        IS000580
00247            DO 130 K=1,LNCAT                                               IS000590
00252            DN(K)=N(K)                                                     IS000600
00253            DO 130 J=1,NOFEAT                                              IS000610
00256            STDEV(J,K )=SQRT(AVP(J,K )-AMN(J,K )**2)                       IS000620
00257        130 MEANS(J,K)=AMN(J,K)                                            IS000630
00257      C*                                                                   IS000640
00257      C*    CALCULATE DISTANCES BETWEEN CLUSTER CENTERS                    IS000650
00257      C*                                                                   IS000660
00262            CALL CLDIST                                                    IS000670
00262      C*                                                                   IS000680
00262      C*    IF ISTOP EQUALS ZERO RETURN AFTER INITIAL CLASSIFICATION       IS000690
00262      C*                                                                   IS000700
00263            IF (MOD(KKT,MAF)) 150,140,150                                  IS000710
00266        140 CALL PRINT(KKT,IPLACE)                                         IS000720
00267            GO TO 170                                                      IS000730
00270        150 IF (MOD(KKT,KRN)) 170,160,170                                  IS000740
00273        160 CONTINUE                                                       IS000750
00274            CALL PRINT(KKT,IPLACE)                                         IS000760
00275        170 IF(ISTOP.EQ.0)RETURN                                           IS000770
00275      C                                                                    IS000780
00275      C     REMOVE CLUSTERS WITH TOO FEW ELEMENTS                          IS000790
00275      C                                                                    IS000800
00277            DO 180 K=1,INCAT                                               IS000810
00302            IF (N(K)-NMIN) 190,180,180                                     IS000820
00305        180 CONTINUE                                                       IS000830
00307            CALL CLDIST                                                    IS000840
00310            GO TO 220                                                      IS000850
00311        190 IF (MOD(KKT,KRN)) 200,,200                                     IS000860
00314            WRITE (6,210)K,N(K),NMIN                                       IS000870
00321        200 CALL DELETE(K)                                                 IS000880
00322            GO TO 170                                                      IS000890
00323        210 FORMAT('0 CLUSTER ',I2,' REMOVED FOR HAVING  ONLY ',I6,        IS000900
00323           2       ' ELEMENTS. MIN. NO. ELEMENTS IS ',I6)                  IS000910
00324        220 CONTINUE                                                       IS000920
```

```
00324   C                                                                         IS000930
00325         IF (KKT.EQ.ISTOP) GO TO 230                                         IS000940
00327         GO TO ITER(230,410)                                                 IS000950
00327   C                                                                         IS000960
00330                                                                             IS000970
00330   C*    SPLIT ITERATION                                                     IS000980
00330   C                                                                         IS000990
00330   C                                                                         IS001000
00330     230 TRIG1=0                                                             IS001010
00331         ASSIGN 410 TO ITER                                                  IS001020
00332         ISPLT=0                                                             IS001030
00333         DO 260 K=1,INCAT                                                    IS001040
00333   C*                                                                        IS001050
00333   C*    FIND MAXIMUM STANDARD DEVIATION PER CLUSTER                         IS001060
00333   C*                                                                        IS001070
00336         SGMA(K)=STDEV(1,K)                                                  IS001080
00337         ISGMA(K)=1                                                          IS001090
00340         DO 250 J=2,KDIM                                                     IS001100
00343         IF (STDEV(J,K)-SGMA(K)) 250,240,240                                 IS001110
00346     240 ISGMA(K)=J                                                          IS001120
00347         SGMA(K)=STDEV(J,K)                                                  IS001130
00350     250 CONTINUE                                                            IS001140
00352         IF(SGMA(K).GE.STDMAX)ISPLT=ISPLT+1                                  IS001150
00354     260 CONTINUE                                                            IS001160
00356         TEST=FLOAT(ISPLT)/FLOAT(LNCAT)                                      IS001170
00357         IF(TEST.LT..20).AND.TRIG2.EQ.0.AND.KKT.NE.ISTOP)GO TO 410
00361         IF (TRIG2.EQ.0) ASSIGN 230 TO ITER                                  IS001190
00361   C                                                                         IS001200
00361   C     IS SPLITTING REQUIRED                                               IS001210
00361   C                                                                         IS001220
00363     270 K=1                                                                 IS001230
00364         NCAT=INCAT                                                          IS001240
00365     280 IF (K-NCAT) 290,290,500                                             IS001250
00370     290 IF (STDMAX-SGMA(K)) 300,300,310                                     IS001260
00373     300 IF (N(K)-(NMIN+NMIN+2)) 310,310,320                                 IS001270
00376     310 K=K+1                                                               IS001280
00377         GO TO 280                                                           IS001290
00377   C                                                                         IS001300
00377   C     SPLIT CLUSTER K                                                     IS001310
00377   C                                                                         IS001320
00400     320 TRIG1=1                                                             IS001330
00401         KX=ISGMA(K)                                                         IS001340
00402     330 INCAT=INCAT+1                                                       IS001350
00403         IF (INCAT.LE.MAXCLS) GO TO 350                                      IS001360
00405         WRITE (6,340)KKT                                                    IS001370
00410     340 FORMAT(/' MAXIMUM CLUSTERS ON ITERATION',I4/' SPLITTING REQUIRED BIS001380
00410        *UT NOT PERFORMED'/)                                                 IS001390
00411         LNCAT=MAXCLS                                                        IS001400
00412         GO TO 500                                                           IS001410
00413     350 INC=INCAT                                                           IS001420
00414         GO TO 360                                                           IS001430
00415     360 DO 370 I=1,KDIM                                                     IS001440
00420     370 AMN(I,INC)=AMN(I,K)                                                 IS001450
00422         IF (SPTRIG.GT.0) GO TO 380                                          IS001460
00424         SEP=SGMA(K)                                                         IS001470
00425     380 AMN(KX,K)=AMN(KX,K)+SEP                                             IS001480
00426         AMN(KX,INC)=AMN(KX,INC)-SEP                                         IS001490
00427         SGMA(K)=0.0                                                         IS001500
```

```
00430          IF (MOD(KKT,KRN)) 400,,400                                             ISO01510
00433          WRITE (6,390)K,KX,INC                                                  ISO01520
00440      390 FORMAT('0 CLUSTER ',I2,' IS SPLIT IN THE ',I2,'TH PARAMETER INTO CISO01530
00440         2LUSTER ',I2)                                                           ISO01540
00441      400 CONTINUE                                                               ISO01550
00442          K=K+1                                                                  ISO01560
00443          GO TO 280                                                              ISO01570
00443    C                                                                            ISO01580
00443    C     EVEN ITERATION                                                         ISO01590
00443    C                                                                            ISO01600
00443    C     ARE CLUSTERS TO BE COMBINED                                            ISO01610
00443    C                                                                            ISO01620
00444      410 CONTINUE                                                               ISO01630
00445          TRIG2=1                                                                ISO01640
00446          ASSIGN 230 TO ITER                                                     ISO01650
00447          DO 420 I=1,INCAT                                                       ISO01660
00452          IP(I)=I                                                                ISO01670
00453      420 CONTINUE                                                               ISO01680
00455          I=0                                                                    ISO01690
00456      430 I=I+1                                                                  ISO01700
00457          IF (I.GE.INCAT) GO TO 470                                              ISO01710
00461          IF (IP(I).NE.I) GO TO 430                                              ISO01720
00463          DMIN=DLMIN                                                             ISO01730
00464          M=I+1                                                                  ISO01740
00465          KK=0                                                                   ISO01750
00466          DO 440 J=M,INCAT                                                       ISO01760
00471          IF (IP(J).NE.J) GO TO 440                                              ISO01770
00473          IF (CLD(I,J).GT.DMIN) GO TO 440                                        ISO01780
00475          DMIN=CLD(I,J)                                                          ISO01790
00476          KK=J                                                                   ISO01800
00477      440 CONTINUE                                                               ISO01810
00501          IF (KK.EQ.0) GO TO 430                                                 ISO01820
00503          IP(KK)=I                                                               ISO01830
00503    C                                                                            ISO01840
00503    C     COMBINE CLUSTERS I AND KK                                              ISO01850
00503    C                                                                            ISO01860
00504      450 CONTINUE                                                               ISO01870
00505          DO 460 K=1,KDIM                                                        ISO01880
00510      460 AMN(K,I)=(DN(I)*AMN(K,I)+DN(KK)*AMN(K,KK))/(DN(I)+DN(KK))              ISO01890
00512          DN(I)=DN(I)+DN(KK)                                                     ISO01900
00513          N(I)=N(I)+N(KK)                                                        ISO01910
00514          IF (MOD(KKT,KRN)) 430,,430                                             ISO01920
00517          WRITE (6,490)I,KK,I                                                    ISO01930
00524          GO TO 430                                                              ISO01940
00525      470 DO 480 I=1,INCAT                                                       ISO01950
00530          IF(IP(I).NE.I)CALL DELETE(I)                                           ISO01960
00532      480 CONTINUE                                                               ISO01970
00534      490 FORMAT('  CLUSTERS ',I2,' AND ',I2,' HAVE BEEN COMBINED INTO CLUSTISO01980
00534         2ER ',I2)                                                               ISO01990
00534    C*                                                                           ISO02000
00534    C*    REINITIALIZE                                                           ISO02010
00534    C*                                                                           ISO02020
00535      500 CONTINUE                                                               ISO02030
00536          DO 510 J=1,MAXCLS                                                      ISO02040
00541          N(J)=0                                                                 ISO02050
00542          DN(J)=0.0                                                              ISO02060
00543          SGMA(J)=0.0                                                            ISO02070
00544          ISGMA(J)=0                                                             ISO02080
```

```
00545    215*            DO 510 K=1,KDIM                                          ISOD2090
00550    216*            AVP(K,J)=0.0                                             ISOD2100
00551    217*            STDEV(K,J)=0.0                                           ISOD2110
00552    218*            MEANS(K,J)=AMN(K,J)                                      ISOD2120
00553    219*            AMN(K,J)=0.0                                             ISOD2130
00554    220*        510 CONTINUE                                                 ISOD2140
00557    221*            IF (KKT.EQ.ISTOP) GO TO 520                              ISOD2150
00561    222*            KKT=KKT+1                                                ISOD2160
00562    223*            GO TO 10                                                 ISOD2170
00562    224*      C                                                              ISOD2180
00562    225*      C     EXIT .                                                   ISOD2190
00562    226*      C                                                              ISOD2200
00563    227*        520 ISTOP=0                                                  ISOD2210
00564    228*            GO TO 10                                                 ISOD2220
00565    229*        530 IF (KKT.NE.2) GO TO 550                                  ISOD2230
00567    230*            WRITE (6,540)                                            ISOD2240
00571    231*        540 FORMAT(' THE ORIGINAL CLUSTER WAS NOT SPLIT - EXAMINE THE INPUT VAISOD2250
00571    232*           *LUE FOR STDMAX'/)                                        ISOD2260
00572    233*            RETURN                                                   ISOD2270
00573    234*        550 WRITE (6,560)KKT                                         ISOD2280
00576    235*        560 FORMAT(//' AFTER ',I4,' ITERATIONS ALL DATA HAS BEEN ASSIGNED TO OISOD2290
00576    236*           *NE CLUSTER'/)                                            ISOD2300
00577    237*            RETURN                                                   ISOD2310
00577    238*      C*                                                             ISOD2320
00577    239*      C*    INTERNAL SUBROUTINE TO DELETE A CLUSTER                  ISOD2330
00577    240*      C*                                                             ISOD2340
00600    241*            SUBROUTINE DELETE(LK                                     ISOD2350
00603    242*            INCAT=INCAT-1                                            ISOD2360
00604    243*            IF(LK.EQ.(INCAT+1))RETURN                                ISOD2370
00606    244*            DO 560 J=LK,INCAT                                        ISOD2380
00611    245*            DO 550 L=1,KDIM                                          ISOD2390
00614    246*            AMN(L,J)=AMN(L,J+1)                                      ISOD2400
00615    247*            MEANS(L,J)=MEANS(L,J+1)
00616    248*        550 STDEV(L,J)=STDEV(L,J+1)                                  ISOD2410
00620    249*            N(J)=N(J+1)                                              ISOD2420
00621    250*        560 DN(J)=DN(J+1)                                            ISOD2430
00623    251*            RETURN                                                   ISOD2440
00624    252*            END                                                      ISOD2450


   END OF COMPILATION:          NO  DIAGNOSTICS.
 ISODAT          SYMBOLIC                           29 NOV 73  13:41:01   0  01561712    14   252  (DELETED)
 ISODAT  CODE    RELOCATABLE                        29 NOV 73  13:41:01   1  01570622    36     1  (DELETED)
                                                                          0  01570666    14    89
```

```
@ FOR,* FIND12,FIND12                                                                                03 DEC 73                12:22:52.319
UNIVAC 1108 FORTRAN V EXEC II LEVEL 25A  -(EXEC8 LEVEL E12010010A)
THIS COMPILATION WAS DONE ON 03 DEC 73 AT 12:22:52


    FUNCTION FIND         ENTRY POINT 000265
              NXTCHR      ENTRY POINT 000311
              NUMBER      ENTRY POINT 000333


    STORAGE USED: CODE(1) 000357; DATA(0) 000041; BLANK COMMON(2) 000000


    EXTERNAL REFERENCES (BLOCK, NAME)

     0003    NERR3S


    STORAGE ASSIGNMENT  (BLOCK, TYPE, RELATIVE LOCATION, NAME)

     0001     000023 115G         0001     000027 120G         0001     000073 140G         0001     000134 160G         0001     000147 184G
     0001     000047 20L          0001     000107 40L          0001     000111 50L          0001     000205 60L          0001     000213 70L
     0001     000224 90L          0000 I 000003 BLANK         0000 I 000004 COMMA         0000 I 000001 CRDSIZ        0000 I 000000 FIND
     0000 I 000007 I              0000     000015 INJPS       0000 I 000011 J             0000 I 000005 K             0000 I 000006 L
     0000 I 000012 NUM            0000 I 000013 NWORD         0000 I 000002 VECSIZ        0000 I 000010 VK


00100         1*      C       //SYNTAX                                                              FIND0000
00101         2*              FUNCTION FIND(CARD,COL,VECTOR)                                        FIND0010
00101         3*      C                                                                             FIND0020
00103         4*              IMPLICIT INTEGER (A-H,O-Z)                                            FIND0030
00103         5*      CI-------------------------------------------------------------------------IFIND0040
00103         6*      CI                                                                           IFIND0050
00103         7*      CI-------------------------------------------------------------------------IFIND0060
00103         8*      CI                                                                           IFIND0070
00103         9*      CI      CALL..     J = FIND(CARD,COL,VECTOR)                                 IFIND0080
00103        10*      CI                                                                           IFIND0090
00103        11*      CI      ARGS..     CARD   - BCD BUFFER                                       IFIND0100
00103        12*      CI                 COL    - PTR TO POSTION IN 'CARD'                         IFIND0110
00103        13*      CI                 VECTOR - VEC CONTAINING N SYMBOLS                         IFIND0120
00103        14*      CI                          TO BE LOCATED IN CARD                            IFIND0130
00103        15*      CI                          (N IS GIVEN IN VECTOR(1)                         IFIND0140
00103        16*      CI                          EG,    /2,'S','*'/       )                       IFIND0150
00103        17*      CI                 NUMVEC - FULL WORD INTEGER VECTOR                         IFIND0160
00103        18*      CI                                                                           IFIND0170
00103        19*      CI      REQUIRES. NCR                                                        IFIND0180
00103        20*      CI                                                                           IFIND0190
00103        21*      CI      PURPOSE.. USED TO LOCATE SPECIAL SYMBOLS IN 'CARD'                   IFIND0200
00103        22*      CI                                                                           IFIND0210
00103        23*      CI      RETURNS.. COL     - PTS AT SYMBOL ( IF LOCATED)                      IFIND0220
00103        24*      CI                J       - PTS AT SYMBOL LOCATED IN 'VECTOR'                IFIND0230
00103        25*      CI                                                                           IFIND0240
00103        26*      CI-------------------------------------------------------------------------IFIND0250
00103        27*      CI                                                                           IFIND0260
00103        28*      CI      CALL..     J = NXTCHR(CARD,COL)                                      IFIND0270
00103        29*      CI                                                                           IFIND0280
```

```
00103    CI   ARGS..     SEE ABOVE                                                    IFIND0290
00103    CI                                                                           IFIND0300
00103    CI   PURPOSE.. LOCATES THE NEXT NONBLANK SYMBOL IN 'CARD'                     IFIND0310
00103    CI                                                                           IFIND0320
00103    CI   RETURNS.. J        - LOCATED CHARACTER (BLANK IF EOC)                   IFIND0330
00103    CI             COL      - SEE ABOVE                                          IFIND0340
00103    CI                                                                           IFIND0350
00103    CI---------------------------------------------------------------------------IFIND0360
00103    CI                                                                           IFIND0370
00103    CI   CALL..    J = NUMBER(CARD,COL,NUMVEC)                                    IFIND0380
00103    CI                                                                           IFIND0390
00103    CI   ARGS..    SEE ABOVE                                                      IFIND0400
00103    CI                                                                           IFIND0410
00103    CI   PURPOSE.. DECODES INTEGERS SFPARATED BY COMMAS                           IFIND0420
00103    CI                       STOPS WHEN NONUMERIC IS FOUND                        IFIND0430
00103    CI                                                                           IFIND0440
00103    CI   RETURNS.. J       - NO OF INTEGERS LOCATED                               IFIND0450
00103    CI             NUMVEC  - CONTAINS THE 'J' INTEGERS                            IFIND0460
00103    CI                                                                           IFIND0470
00103    CI---------------------------------------------------------------------------IFIND0480
00103    CI---------------------------------------------------------------------------IFIND0490
00103    C                                                                             FIND0500
00103    C                                                                             FIND0510
00103    C                                                                             FIND0520
00104          DIMENSION NUMVEC(1),CARD(1),VECTOR(1)                                   FIND0530
00104    C                                                                             FIND0540
00105          DATA CRDSIZ/62/, VECSIZ/30/, BLANK/' '/, COMMA/','/                     FIND0550
00105    C                                                                             FIND0560
00105    C     -------------------------------------------------------------------------FIND0570
00105    C     -------------------------------------------------------------------------FIND0580
00105    C                                                                             FIND0590
00112          K = VECTOR(1)+1                                                         FIND0600
00113          L = COL+1                                                               FIND0610
00114          DO 10 COL=L,CRDSIZ                                                      FIND0620
00117          DO 10 I=2,K                                                             FIND0630
00122       10 IF (CARD(COL).EQ.VECTOR(I)) GO TO 20                                    FIND0640
00126          I =-1                                                                   FIND0650
00127          COL = L-1                                                               FIND0660
00130       20 FIND =I                                                                 FIND0670
00130    C     WRITE( 6,102) (CARD(K),K=1,62),COL,I,VECTOR(I)                          FIND0680
0013     C 102 FORMAT(' FIND ENTERED'/' ',62A1,I10/' ',I5,A4)                          FIND0690
00131          RETURN                                                                  FIND0700
00131    C                                                                             FIND0710
00131    C     -------------------------------------------------------------------------FIND0720
00131    C                                                                             FIND0730
00132          ENTRY NXTCHR(CARD,COL)                                                  FIND0740
00134          L = COL+1                                                               FIND0750
00135          IF (L.GT.CRDSIZ) GO TO 40                                               FIND0760
00137          DO 30 COL=L,CRDSIZ                                                      FIND0770
00142          FIND = CARD(COL)           @ *****                                      FIND0780
00143          IF (FIND.NE.BLANK) GO TO 50         @ *****                             FIND0790
00145       30 CONTINUE                                                                FIND0800
00147          COL=CRDSIZ-1                                                            FIND0810
00150       40 FIND = BLANK               @ *****                                      FIND0820
00151       50 CONTINUE                                                                FIND0830
00151    C     WRITE (6,104) (CARD(K),K=1,62),COL,NXTCHR                               FIND0840
00151    C 104 FORMAT(' NXTCHR ENTERED'/' ',62A1,I10/' 'A4)                            FIND0850
00152          RETURN                                                                  FIND0860
```

```
00152     88*     C                                                                    FIND0870
00152     89*     C     ---------------------------------------------------------------FIND0880
00152     90*     C                                                                    FIND0890
00152     91*     C                                                                    FIND0900
00153     92*           ENTRY NUMBER(CARD,COL,NUMVEC)                                  FIND0910
00155     93*           L = COL+1                                                      FIND0920
00156     94*           VK=VECSIZ                                                      FIND0930
00157     95*           DO 80 J=1,VK                                                   FIND0940
00162     96*           NUM = 0                                                        FIND0950
00163     97*           DO 60 COL=L,CRDSIZ                                             FIND0960
00166     98*           IF (CARD(COL).EQ.BLANK) GO TO 60                               FIND0970
00170     99*           IF (CARD(COL).EQ.COMMA) GO TO 70                               FIND0980
00172    100*           NWORD = FLD (ABS(0), 6, CARD(COL))                             FIND0990
00173    101*           NWORD = NWORD - 48                                             FIND1000
00174    102*           IF (NWORD.LT.0.OR.NWORD.GT.9) GO TO 90                         FIND1010
00176    103*           NUM = 10 * NUM + NWORD                                         FIND1020
00177    104*        60 CONTINUE                                                       FIND1030
00201    105*           COL = CRDSIZ+1                                                 FIND1040
00202    106*           GO TO 90                                                       FIND1050
00203    107*        70 NUMVEC(J) = NUM                                                FIND1060
00204    108*           L = COL+1                                                      FIND1070
00205    109*        80 CONTINUE                                                       FIND1080
00207    110*           J = VECSIZ                                                     FIND1090
00210    111*        90 COL = COL-1                                                    FIND1100
00211    112*           NUMVEC(J) = NUM                                                FIND1110
00212    113*           FIND = J       @ *****                                         FIND1120
00212    114*     C     WRITE( 6,106) (CARD(K),K=1,62),COL,NUMBER,(NUMVEC(K),K=1,J)    FIND1130
00212    115*     C 106 FORMAT(' NUMBER ENTERED'/' '62A1,I10/' ' ,I5,18I3)             FIND1140
00213    116*           RETURN                                                         FIND1150
00213    117*     C*****                                                               FIND1160
00213    118*     C*****   FUNCTION ENTRIES MUST RETURN VALUE IN ORIGINAL FUNCTION NAME   FIND1170
00213    119*     C*****                                                               FIND1180
00214    120*             END                                                          FIND1190


      END OF COMPILATION:          NO  DIAGNOSTICS.
    FIND12          SYMBOLIC                                      29 NOV 73  13:40:55   0  01547436    14   120  (DELETED)
    FIND12  CODE    RELOCATABLE                                   29 NOV 73  13:40:55   1  01552656    24     1  (DELETED)
                                                                                        0  01552706    14    25
```

```
@ FOR,* FLTNUM,FLTNUM                                                                                03 DEC 73          12:22:54.220
UNIVAC 1108 FORTRAN V EXEC II LEVEL 25A  -(EXEC8 LEVEL E12010010A)
THIS COMPILATION WAS DONE ON 03 DEC 73 AT 12:22:54


   FUNCTION FLTNUM        ENTRY POINT 000302


   STORAGE USED: CODE(1) 000326; DATA(0) 000047; BLANK COMMON(2) 000000


   EXTERNAL REFERENCES (BLOCK, NAME)

    0003    NEXP5$
    0004    NERR3$


   STORAGE ASSIGNMENT  (BLOCK, TYPE, RELATIVE LOCATION, NAME)

    0001     000013 10L         0001     000263 110L       0001     000032 130G       0001     000056 20L        0001     000204 202G
    0001     000260 220G        0001     000074 30L        0001     000103 40L        0001     000135 50L        0001     000141 60L
    0001     000150 70L         0001     000222 90L        0000 I 000003 BLANK      0000 I 000004 COMMA      0000 I 000012 CRDSIZ
    0000 I 000001 FLTNUM      0000 I 000023 I          0000     000026 INJP$     0000 I 000017 ITER       0000 I 000013 L
    0000 I 000006 MINUS       0000 I 000021 MORNUM     0000 I 000016 PCNT       0000 I 000010 PERIOD     0000 I 000005 PLUS
    0000 R 000002 PNUM        0000 I 000020 SIDE       0000 I 000000 SIGN       0000 I 000007 STAR       0000 I 000022 VECFIN
    0000 I 000014 VECPOS      0000 I 000015 NNUM       0000 I 000011 ZERO



00100        1*      C      //FLTNUM                                                        FLTN0000
00101        2*             FUNCTION FLTNUM(CARD,COL,NUMVEC,VECMAX)                         FLTN0010
00101        3*      C                                                                      FLTN0020
00103        4*             IMPLICIT INTEGER (A-H,O-Z)                                      FLTN0030
00103        5*      CI---------------------------------------------------------------------IFLTN0040
00103        6*      CI---------------------------------------------------------------------IFLTN0050
00103        7*      CI                                                                     IFLTN0060
00103        8*      CI     CALL..     J = FLTNUM(CARD,COL,NUMVEC,VECMAX)                    IFLTN0070
00103        9*      CI                                                                     IFLTN0080
00103       10*      CI     ARGS..     CARD   - 62 COL CARD BUFFER                            IFLTN0090
00103       11*      CI                COL    - PTR TO FIRST COL IN CARD TO SCAN              IFLTN0100
00103       12*      CI                NUMVEC - BUFFER IN WHICH TO RETURN THE NUMBERS         IFLTN0110
00103       13*      CI                VECSIZ - LENGTH OF NUMVEC                              IFLTN0120
00103       14*      CI                                                                     IFLTN0130
00103       15*      CI     REQUIRES. NONE                                                  IFLTN0140
00103       16*      CI                                                                     IFLTN0150
00103       17*      CI     PURPOSE.. INTERPRETS REAL NUMBERS SEPARTED BY COMMAS ON CARD    IFLTN0160
00103       18*      CI               AND RETURNS THEM IN NUMVEC.                           IFLTN0170
00103       19*      CI                           STOPS AT FIRST 'NONUMERIC'                IFLTN0180
00103       20*      CI                           (NOTE. NUMBERS MAY APPEAR IN              IFLTN0190
00103       21*      CI                                  'DATA STATEMENT FORMAT' )          IFLTN0200
00103       22*      CI                                                                     IFLTN0210
00103       23*      CI     RETURNS.. COL    -  COLUMN WHERE SCAN TERMINATED                IFLTN0220
00103       24*      CI               NUMVEC -  VECTOR OF REAL NUMBERS FOUND                IFLTN0230
00103       25*      CI               FLTNUM -  NO OF REAL NUMBERS RETURNED                 IFLTN0240
00103       26*      CI                                                                     IFLTN0250
00103       27*      CI---------------------------------------------------------------------IFLTN0260
00103       28*      CI---------------------------------------------------------------------IFLTN0270
```

ORIGINAL PAGE IS
OF POOR QUALITY

```
00103     C                                                                      FLTN0280
00103     C                                                                      FLTN0290
00104           REAL NUMVEC(20),PNUM                                             FLTN0300
00105           DIMENSION CARD(62)                                               FLTN0310
00105           DATA BLANK/' '/,COMMA/','/,PLUS/'+'/,MINUS/'-'/,STAR/'*'/,       FLTN0320
00106          1     PERIOD/'.'/, ZERO/'0'/, CRDSIZ/62/                           FLTN0330
00106     C                                                                      FLTN0340
00106     C     ----------------------------------------------------------------FLTN0350
00106     C                                                                      FLTN0360
00117           L = COL+1                                                        FLTN0370
00120           VECPOS = 1                                                       FLTN0380
00121        10 WNUM = 0                                                         FLTN0390
00122           PCNT = 0                                                         FLTN0400
00123           PNUM = 0.0                                                       FLTN0410
00124           ITER = 1                                                         FLTN0420
00125           SIDE = -1                                                        FLTN0430
00126           SIGN = +1                                                        FLTN0440
00126     C                                                                      FLTN0450
00127           DO 60 COL=L,CRDSIZ                                               FLTN0460
00132           IF (CARD(COL).EQ.BLANK) GO TO 60                                 FLTN0470
00134           IF (CARD(COL).EQ.PLUS) GO TO 60                                  FLTN0480
00136           IF (CARD(COL).EQ.COMMA) GO TO 70                                 FLTN0490
00140           IF (CARD(COL).NE.MINUS) GO TO 20                                 FLTN0500
00142           SIGN = -SIGN                                                     FLTN0510
00143           GO TO 60                                                         FLTN0520
00144        20 IF (CARD(COL).NE.STAR) GO TO 30                                  FLTN0530
00146           ITER = WNUM                                                      FLTN0540
00147           WNUM = 0                                                         FLTN0550
00150           PCNT = 0                                                         FLTN0560
00151           PNUM = 0.0                                                       FLTN0570
00152           SIDE = -1                                                        FLTN0580
00153           SIGN = +1                                                        FLTN0590
00154           GO TO 60                                                         FLTN0600
00155        30 IF (CARD(COL).NE.PERIOD) GO TO 40                                FLTN0610
00157           SIDE = 1                                                         FLTN0620
00160           GO TO 60                                                         FLTN0630
00161        40 IF (CARD(COL).LT.ZERO) GO TO 90                                  FLTN0640
00163           MORNUM = FLD(ABS(0), 6, CARD(COL)) - 48                          FLTN0650
00164           IF (SIDE.LT.0) GO TO 50                                          FLTN0660
00166           PCNT = PCNT+1                                                    FLTN0670
00167           PNUM = PNUM+MORNUM*(0.1**PCNT)                                   FLTN0680
00170           GO TO 60                                                         FLTN0690
00171        50 WNUM = 10*WNUM+MORNUM                                            FLTN0700
00172        60 CONTINUE                                                         FLTN0710
00172     C                                                                      FLTN0720
00174           COL = CRDSIZ+1                                                   FLTN0730
00175           GO TO 90                                                         FLTN0740
00176        70 VECFIN = VECPOS+ITER-1                                           FLTN0750
00177           IF ( VECFIN .GT. VECMAX ) VECFIN = VECMAX-1                      FLTN0760
00201           DO 80 I=VECPOS,VECFIN                                            FLTN0770
00204        80 NUMVEC(I) = SIGN*(WNUM+PNUM)                                     FLTN0780
00206           L = COL+1                                                        FLTN0790
00207           VECPOS = VECFIN+1                                                FLTN0800
00210           IF (VECPOS.LE.VECMAX) GO TO 10                                   FLTN0810
00212           GO TO 110                                                        FLTN0820
00213        90 COL = COL-1                                                      FLTN0830
00214           VECFIN = VECPOS+ITER-1                                           FLTN0840
00215           IF ( VECFIN .GT. VECMAX ) VECFIN = VECMAX                        FLTN0850
```

```
00217     87*            DO 100 I=VECPOS,VECFIN                                   FLTN0860
00222     88*        100 NUMVEC(I) = SIGN*(WNUM+PNUM)                             FLTN0870
00224     89*        110 FLTNUM = VECFIN                                          FLTN0880
00224     90*      C     WRITE(6,706)  (CARD(K),K=1,62), COL,FLTNUM,(NUMVEC(K),K=1,FLTNUM) FLTN0890
00224     91*      C 706 FORMAT(' FLTNUM ENTERED'/' ', 62A1,I10/' ',I5,20F8.2,/' ',10F8.2) FLTN0900
00225     92*            RETURN                                                   FLTN0910
00226     93*             END                                                     FLTN0920


      END OF COMPILATION:        NO  DIAGNOSTICS. .
   FLTNUM          SYMBOLIC                           29 NOV 73  13:40:56   0  01553444    14    93  (DELETED)
   FLTNUM  CODE    RELOCATABLE                        29 NOV 73  13:40:56   1  01556072    24     1  (DELETED)
                                                                            0  01556122    14    23
```

```
@ FOR,* PACK,PACK                                                                              03 DEC 73              12:23: 1.653
UNIVAC 1108 FORTRAN V EXEC II LEVEL 25A  -(EXEC8 LEVEL E12010010A)
THIS COMPILATION WAS DONE ON 03 DEC 73 AT 12:23:01

   SUBROUTINE PACK       ENTRY POINT 000206

   STORAGE USED: CODE(1) 000237; DATA(0) 000025; BLANK COMMON(2) 000000

   EXTERNAL REFERENCES (BLOCK, NAME)

    0003   NERR3$

   STORAGE ASSIGNMENT  (BLOCK, TYPE, RELATIVE LOCATION, NAME)

    0001   000026 10L        0001   000105 20L        0000 I 000000 IBIT      0000   000006 INJP$      0000 I 000005 JBIT
    0000 I 000001 OBIT       0000 I 000004 OVER       0000 I 000003 REMAIN    0000 I 000002 WDIN


00101      1*           SUBROUTINE PACK( LNGTH,BIAS,NSAMP,INBUF,OBUF,WDOUT)                    PACK0000
00103      2*           IMPLICIT INTEGER(A-Z)                                                   PACK0010
00104      3*           DIMENSION INBUF(1),OBUF(1)                                              PACK0020
00105      4*           IBIT=36-LNGTH                                                           PACK0030
00106      5*           WDOUT=1+BIAS/36                                                         PACK0040
00107      6*           OBIT=IABS(BIAS-(BIAS/36)*36)
00110      7*           WDIN = 1                                                                PACK0060
00111      8*        10 IF(WDIN.GT.NSAMP)RETURN                                                 PACK0070
00113      9*           IF ((OBIT+LNGTH).GT.36) GO TO 20                                        PACK0080
00115     10*           FLD(OBIT,LNGTH,OBUF(WDOUT))=FLD(IBIT,LNGTH,INBUF(WDIN))                PACK0090
00116     11*           WDIN=WDIN+1                                                             PACK0100
00117     12*           OBIT=OBIT+LNGTH                                                         PACK0110
00120     13*           IF (OBIT.LT.36) GO TO 10                                                PACK0120
00122     14*           OBIT=0                                                                  PACK0130
00123     15*            WDOUT=WDOUT+1                                                          PACK0140
00124     16*           GO TO 10                                                                PACK0150
00125     17*        20 REMAIN=36-OBIT                                                          PACK0160
00126     18*           OVER=LNGTH-REMAIN                                                       PACK0170
00127     19*           FLD(OBIT,REMAIN,OBUF(WDOUT))=FLD(IBIT,REMAIN,INBUF(WDIN))              PACK0180
00130     20*           WDOUT=WDOUT+1                                                           PACK0190
00131     21*           JBIT=IBIT+REMAIN                                                        PACK0200
00132     22*           OBIT=0                                                                  PACK0210
00133     23*           FLD(OBIT,OVER,OBUF(WDOUT))=FLD(JBIT,OVER,INBUF(WDIN))                  PACK0220
00134     24*           WDIN=WDIN+1                                                             PACK0230
00135     25*           OBIT=OBIT+OVER                                                          PACK0240
00136     26*           GO TO 10                                                                PACK0250
00137     27*           END                                                                     PACK0260

     END OF COMPILATION:          NO  DIAGNOSTICS.
   PACK           SYMBOLIC                                       29 NOV 73   13:41:02   0  01573224    14     27  (DELETED)
   PACK     CODE  RELOCATABLE                                    29 NOV 73   13:41:02   1  01574016    24      1  (DELETED)
                                                                                        0  01574056    14     17
```

C-2

```
@ FOR,* PCHSTA,PCHSTA                                                                                03 DEC 73              12:23: 2*93
UNIVAC 1108 FORTRAN V EXEC II LEVEL 25A  -(EXEC8 LEVEL E12010010A)
THIS COMPILATION WAS DONE ON 03 DEC 73 AT 12:23:03


   SUBROUTINE PCHSTA     ENTRY POINT 000365


   STORAGE USED: CODE(1) 000600; DATA(0) 000200; BLANK COMMON(2) 000000

    COMMON BLOCKS:

    0003     PASS    026574


   EXTERNAL REFERENCES (BLOCK, NAME)

    0004     NNCOD$
    0005     NWDU$
    0006     NI02$
    0007     NI01$
    0010     NERR3$


   STORAGE ASSIGNMENT  (BLOCK, TYPE, RELATIVE LOCATION, NAME)

    0000     000054 100F        0000     000057 110F        0000     000066 120F        0000     000077 130F        0000     000102 140F
    0000     000107 150F        0001     000060 157G        0000     000114 160F        0001     000072 165G        0000     000117 170F
    0001     000077 170G        0001     000115 200G        0001     000136 210G        0001     000150 216G        0001     000156 223G
    0001     000163 227G        0001     000200 236G        0001     000205 242G        0001     000230 254G        0001     000246 240G
    0001     000266 267G        0001     000300 276G        0001     000224 30L         0001     000320 302G        0001     000340 311G
    0001     000421 344G        0001     000442 353G        0001     000513 40L         0000     000041 60F         0000     000044 70F
    0000     000047 80F         0000     000051 90F         0000 R   000011 BLANK       0003 I   007305 BLK         0003 I   006155 BLOCK
    0003 R   026507 CHNTHS      0000 R   000004 CLASS       0003 R   007706 CLD         0003 I   007677 DAS         0003 R   000116 DLMIN
    0000 R   000025 E           0003 I   000053 FETVEC      0003 I   006011 FLDNAM      0003 I   007700 FORMAT      0000 R   000007 FREQ
    0003 I   000000 HEAD        0000 I   000026 I           0000 I   000040 IBBT        0000 I   000036 IBIT        0000 I   000020 IBT
    0000 I   000006 ICAL        0003 I   026510 ICHAIN      0003 I   026506 ICHN        0000 I   000000 IFLDSV      0000     000131 INJP$
    0000     000125 INJP$       0000 I   000034 INUM        0000 I   000012 IPLUS       0003 I   007703 IRD         0000 I   000005 IRUN
    0000 I   000037 ISIGN       0003 I   000111 ISTOP       0000 I   000013 IW          0000 I   000035 IWRD        0000 I   000027 J
    0003 I   007704 KPTS        0003 I   000114 KRN         0003 I   026573 KUNIT       0000 I   000030 L           0000 I   000031 LBLK
    0003 I   000112 LNCAT       0000 I   000033 LREC        0003 I   007701 MAP         0003 I   026505 MAXCLS      0003 R   000120 MEANS
    0003 I   014612 N           0003 I   014674 NBLK        0003 I   000113 NMIN        0003 I   000052 NOFEAT      0003 I   006010 NOFLD
    0003 I   007705 NOPTS       0000 I   000032 NREC        0000 I   000010 PCHUNT      0003 I   026504 PUNCH       0003 R   000117 SEP
    0003 I   007702 SPTRIG      0003 R   003054 STDEV       0003 R   000115 STDMAX      0003 I   007615 SYMBLS      0003 I   026572 VARSIZ


00101          1*              SUBROUTINE PCHSTA(LOCK,COVAR)                                    PCHS0000
00103          2*              INCLUDE COMMON,LIST                                              PCHS0010
00104          2*              PARAMETER MAXPOP=50,MAXFET=30
00105          2*              PARAMETER MAXDIM=25000
00106          2*              PARAMETER MAXFLD=100
00107          2*              COMMON/PASS/HEAD(42),NOFEAT,FETVEC(MAXFET),ISTOP,LNCAT,NMIN,KRN,
00107          2*             * STDMAX,DLMIN,SEP,MEANS(MAXFET,MAXPOP),STDEV(MAXFET,MAXPOP),
00107          2*             * NOFLD,FLDNAM(MAXFLD),BLOCK(MAXFLD,6),BLK(MAXFLD,2),
00107          2*             * SYMBLS(MAXPOP),DAS,FORMAT,MAP,
00107          2*             * SPTRIG,IRD,KPTS,NOPTS,CLD(MAXPOP,MAXPOP)
```

```
00107         *, N(MAXPOP),NBLK(MAXFLD,MAXPOP)
00107         * ,PUNCH,MAXCLS,ICHN,CHNTHS,ICHAIN(MAXPOP),VARSIZ
00107         *,KUNIT
00110          INTEGER VARSIZ
00111          INTEGER PUNCH
00112          REAL MEANS
00113          INTEGER SPTRIG,BLOCK,BLK,SYMBLS,HEAD,FETVEC,FLDNAM,DAS
00114          INTEGER FORMAT
00115          END
00116          DIMENSION IFLDSV(4)
00117          DATA CLASS/'CLASS'/,IRUN/1/,ICAL/0/,IFLDSV/1,25,1,25/,FREQ/1.0/
00125          INTEGER PCHUNT                                                  PCHS0020
00126          DATA PCHUNT/-3/                                                 PCHS0030
00130          DIMENSION COVAR(LNCAT,VARSIZ)                                   PCHS0040
00131          DIMENSION  LOCK(1)                                              PCHS0050
00132          DATA BLANK/1H /,IPLUS/042/                                      PCHS0060
00135          DIMENSION IW(5),IBT(5)                                          PCHS0070
00136          DATA IW/3,5,7,10,12/,IBT/6,18,30,6,18/                          PCHS0080
00141          DATA E/'E'/                                                     PCHS0090
00143          IF (PUNCH.EQ.2) GO TO 30                                        PCHS0100
00145          WRITE(PCHUNT,110)
00147          WRITE(PCHUNT,120)LNCAT,LNCAT,NOFEAT,ICAL
00155          WRITE(PCHUNT,130)(FETVEC(I),I=1,NOFEAT)
00163          WRITE(PCHUNT,140)(IRUN,(IFLDSV(I),I=1,4),J,J=1,LNCAT)
00176          WRITE(PCHUNT,150)(CLASS,J,J=1,LNCAT)
00205          L=2*NOFEAT
00206          WRITE(PCHUNT,160)(FREQ,I=1,L)
00214          WRITE(PCHUNT,170)(N(I),I=1,LNCAT)
00222          DO 10 J=1,LNCAT                                                 PCHS0110
00225          WRITE (PCHUNT,90)(MEANS(I,J),I=1,NOFEAT)                        PCHS0120
00233       10 CONTINUE                                                        PCHS0130
00235          DO 20 J=1,LNCAT                                                 PCHS0140
00240          WRITE (PCHUNT,100)(COVAR(J,I),I=1,VARSIZ)                       PCHS0150
00246       20 CONTINUE                                                        PCHS0160
00250          IF(PUNCH.EQ.1)RETURN                                            PCHS0170
00252       30 CONTINUE                                                        PCHS0180
00253          DO 40 J=1,LNCAT                                                 PCHS0190
00256          ENCODE (60,LOCK)(MEANS(I,J),I=1,NOFEAT)                         PCHS0200
00264          CALL MASK(NOFEAT)                                               PCHS0210
00265          WRITE (PCHUNT,80)(LOCK(I),I=1,LBLK)                             PCHS0220
00273       40 CONTINUE                                                        PCHS0230
00275          DO 50 J=1,LNCAT                                                 PCHS0240
00300          ENCODE (70,LOCK)(COVAR(J,I),I=1,VARSIZ)                         PCHS0250
00306          CALL MASK(VARSIZ)                                               PCHS0260
00307          WRITE (PCHUNT,80)(LOCK(I),I=1,LBLK)                             PCHS0270
00315       50 CONTINUE                                                        PCHS0280
00317          RETURN                                                          PCHS0290
00320       60 FORMAT('MN',5E14.8,8X)                                          PCHS0300
00321       70 FORMAT('CV',5E14.8,8X)                                          PCHS0310
00322       80 FORMAT(13A6,A2)                                                 PCHS0320
00323       90 FORMAT('MEAN ',5E15.8)                                          PCHS0330
00324      100 FORMAT('COVAR',5E15.8)                                          PCHS0340
00325      110 FORMAT('MODULE TRAINING FIELD DECK*$ISOCLS')
00326      120 FORMAT('CLASS',I5,'FIELD',I5,'FEAT',I6,'CAL',I2)
00327      130 FORMAT('VECTR',30I2)
00330      140 FORMAT(I8,4I5,I2,11X,I8,4I5,I2,9X)
00331      150 FORMAT('CLSDESC    ',8(A6,I2))
00332      160 FORMAT('FREQ ',12F6.2)
```

```
00333      52*          170 FORMAT('NOPTS',8I9)
00333      53*       C*                                                                          PCHS0350
00333      54*       C*                                                                          PCHS0360
00333      55*       C*                                                                          PCHS0370
00334      56*              SUBROUTINE MASK(NUMWRD)                                              PCHS0380
00337      57*              NREC=NUMWRD/5                                                        PCHS0390
00340      58*              LREC=NUMWRD-NREC*5                                                   PCHS0400
00341      59*              NREC=NREC+1                                                          PCHS0410
00342      60*              INUM=5                                                               PCHS0420
00343      61*              DO 40 I=1,NREC                                                       PCHS0430
00346      62*              IF(I.EQ.NREC)INUM=LREC                                               PCHS0440
00350      63*              IF(INUM.EQ.0)GO TO 40                                                PCHS0450
00352      64*              DO 30 J=1,INUM                                                       PCHS0460
00355      65*              IWRD=IW(J)+(I-1)*14                                                  PCHS0470
00356      66*              IBIT=IBT(J)                                                          PCHS0480
00357      67*              ISIGN=FLD(IBIT,6, LOCK(IWRD))                                        PCHS0490
00360      68*              IF(ISIGN.EQ.IPLUS)FLD(IBIT,6, LOCK(IWRD))=FLD(0,6,BLANK)             PCHS0500
00362      69*              IBBT=ABS(IBIT-6)                                                     PCHS0510
00363      70*              FLD(IBBT,6,LOCK(IWRD))=FLD(0,6,E)                                    PCHS0520
00364      71*           30 CONTINUE                                                             PCHS0530
00366      72*           40 CONTINUE                                                             PCHS0540
00370      73*              IF(LREC.EQ.0)LBLK=(NREC-1)*14                                        PCHS0550
00372      74*              IF(LREC.GT.0)LBLK=(NREC-1)*14 + IW(LREC)                             PCHS0560
00374      75*              IF(LREC.EQ.3)LBLK=LBLK+1                                             PCHS0570
00376      76*              RETURN                                                               PCHS0580
00377      77*              END                                                                  PCHS0590

     END OF COMPILATION:          NO  DIAGNOSTICS.
    PCHSTA            SYMBOLIC                                          29 NOV 73  13:41:05   0  01574424    14     77  (DELETED)
    PCHSTA  CODE      RELOCATABLE                                       29 NOV 73  13:41:05   1  01576512    24      1  (DELETED)
                                                                                              0  01576542    14     48
```

```
@ FOR,* PRINT,PRINT                                                                    03 DEC 73           12:23: 5.1
UNIVAC 1108 FORTRAN V EXEC II LEVEL 25A  -(EXEC8 LEVEL E1201D010A)
THIS COMPILATION WAS DONE ON 03 DEC 73 AT 12:23:05

    SUBROUTINE PRINT      ENTRY POINT 001006

    STORAGE USED: CODE(1) 001035; DATA(0) 001166; BLANK COMMON(2) 000000

     COMMON BLOCKS:

     0003    PASS    026574

    EXTERNAL REFERENCES (BLOCK, NAME)

     0004    NTRAN
     0005    SETMRG
     0006    NWDU$
     0007    NIO2$
     0010    NIO1$
     0011    NERR3$

    STORAGE ASSIGNMENT  (BLOCK, TYPE, RELATIVE LOCATION, NAME)

     0001    000033 10L       0001    000416 110L      0001    000510 140L      0001    000057 143G      0001    000101 155G
     0001    000657 160L      0001    000700 170L      0001    000127 172G      0001    000137 200G      0001    000146 205G
     0000    000713 210F      0001    000172 217G      0000    000715 220F      0001    000202 225G      0000    000716 230F
     0001    000211 232G      0000    000721 240F      0001    000261 246G      0000    000732 250F      0001    000273 253G
     0000    000741 260F      0001    000302 260G      0000    000747 270F      0000    000757 280F      0000    000763 290F
     0000    000767 300F      0001    000363 303G      0001    000369 306G      0000    000777 310F      0000    001003 320F
     0001    000435 327G      0000    001011 330F      0000    001046 340F      0001    000456 340G      0000    001060 350F
     0000    001063 360F      0001    000567 367G      0000    001064 370F      0001    000575 374G      0000    001103 380F
     0001    000602 400G      0001    000622 411G      0001    000624 414G      0001    000711 441G      0001    000734 452G
     0001    000235 60L       0001    000335 80L       0001    000351 90L       0000 R  000671 BLANK     0003 I  007305 BLK
     0003 I  006155 BLOCK     0003    026507 CHNTHS    0003 R  007706 CLD       0000 I  000001 COL       0003 I  007677 DAS
     0003    000116 DLMIN     0003 I  000053 FETVEC    0003 I  006011 FLDNAM    0003 I  007700 FORMAT    0003 I  000000 HEAD
     0000 I  000674 I         0000 I  000706 IB        0000 I  000700 ICCT      0003    026510 ICHAIN    0003    026506 ICHN
     0000 I  000707 IE        0000 I  000704 IFLD      0000 I  000710 INC       0000    001127 INJP$     0000 I  000677 IRC
     0003 I  007703 IRD       0000 I  000702 IREC      0000 I  000701 ISTAT     0003    000111 ISTOP     0000 I  000673 J
     0000 I  000703 JPTS      0000 I  000676 K         0000 I  000672 KNT       0003 I  007704 KPTS      0003    000114 KRN
     0003    026573 KUNIT     0000 I  000675 L         0000 I  000712 LINE      0000 I  000705 LINES     0003 I  000112 LNCAT
     0000 I  000711 LPTS      0003 I  007701 MAP       0003    026505 MAXCLS    0003 R  000120 MEANS     0003 I  014612 N
     0003 I  014674 NBLK      0003    000113 NMIN      0003 I  000052 NOFEAT    0003 I  006010 NOFLD     0003 I  007705 NOPTS
     0000 I  000513 OUT       0000 I  000000 PTS       0003 I  026504 PUNCH     0003    000117 SEP       0003 I  007702 SPTRIG
     0003 R  003054 STDEV     0003    000115 STDMAX    0003 I  007615 SYMBLS    0003 I  026572 VARSIZ


00101         1*           SUBROUTINE PRINT(KKT,IPLACE)                                          PRIN0000
00103         2*           INCLUDE COMMON,LIST                                                   PRIN0010
00104         2*           PARAMETER MAXPOP=50,MAXFET=30
00105         2*           PARAMETER MAXDIM=25000
00106         2*           PARAMETER MAXFLD=100
00107         2*           COMMON/PASS/HEAD(42),NOFEAT,FETVEC(MAXFET),ISTOP,LNCAT,NMIN,KRN,
00107       2*          • STDMAX,DLMIN,SEP,MEANS(MAXFET,MAXPOP),STDEV(MAXFET,MAXPOP),
00107       2*          • NOFLD,FLDNAM(MAXFLD),BLOCK(MAXFLD,6),BLK(MAXFLD,2),
00107       2*          • SYMBLS(MAXPOP),DAS,FORMAT,MAP,
00107       2*          • SPTRIG,IRD,KPTS,NOPTS,CLD(MAXPOP,MAXPOP)
00107       2*          •, N(MAXPOP),NBLK(MAXFLD,MAXPOP)
00107       2*          • ,PUNCH,MAXCLS,ICHN,CHNTHS,ICHAIN(MAXPOP),VARSIZ
00107       2*          •,KUNIT
00110       2*           INTEGER VARSIZ
00111       2*           INTEGER PUNCH
00112       2*           REAL MEANS
00113       2*           INTEGER SPTRIG,BLOCK,BLK,SYMBLS,HEAD,FETVEC,FLDNAM,DAS
00114       2*           INTEGER FORMAT
00115       2*           END
00116       3*           DIMENSION IPLACE(NOPTS)                                  PRIN0020
00117       4*           INTEGER PTS,COL,OUT                                      PRIN0030
00120       5*           DIMENSION COL(3,110),OUT(110)                            PRIN0040
00121       6*           DATA BLANK/' '/                                          PRIN0050
00123       7*           CALL NTRAN(7,10)                                         PRIN0060
00124       8*           WRITE (6,360)                                            PRIN0070
00126       9*           WRITE (6,HEAD)                                           PRIN0080
00130      10*           IF (KKT.LT.0) GO TO 10                                   PRIN0090
00132      11*           WRITE (6,240)KKT                                         PRIN0100
00135      12*        10 CONTINUE                                                 PRIN0110
00136      13*           WRITE (6,250)LNCAT                                       PRIN0120
00141      14*           KNT=0                                                    PRIN0130
00142      15*           DO 20 J=1,NOFLD                                          PRIN0140
00145      16*        20 KNT=KNT + BLK(J,1)*BLK(J,2)                              PRIN0150
00147      17*           WRITE (6,260)KNT                                         PRIN0160
00152      18*           WRITE (6,270)                                            PRIN0170
00154      19*           DO 30 J=1,LNCAT                                          PRIN0180
00157      20*           WRITE (6,280)J,SYMBLS(J),N(J)                            PRIN0190
00164      21*        30 CONTINUE                                                 PRIN0200
00166      22*           WRITE (6,290)                                            PRIN0210
00170      23*           WRITE (6,300)(BLANK,FETVEC(J),J=1,NOFEAT)                PRIN0220
00177      24*           DO 40 J=1,LNCAT                                          PRIN0230
00202      25*           WRITE (6,310)J,(MEANS(I,J),I=1,NOFEAT) .                 PRIN0240
00211      26*        40 CONTINUE                                                 PRIN0250
00213      27*           WRITE (6,320)                                            PRIN0260
00215      28*           WRITE (6,300)(BLANK,FETVEC(J),J=1,NOFEAT)                PRIN0270
00224      29*           DO 50 J=1,LNCAT                                          PRIN0280
00227      30*           WRITE (6,310)J,(STDEV(I,J),I=1,NOFEAT)                   PRIN0290
00236      31*        50 CONTINUE                                                 PRIN0300
00240      32*           L=1                                                      PRIN0310
00241      33*           J=LNCAT                                                  PRIN0320
00242      34*           IF(J.GT.15)J=15
00244      35*        60 WRITE (6,340)(K,K=L,J)                                   PRIN0340
00252      36*           DO 70 I=1,LNCAT                                          PRIN0350
00255      37*        70 WRITE (6,350)I,(CLD(I,K),K=L,J)                          PRIN0360
00265      38*           IF (J.EQ.LNCAT) GO TO 80                                 PRIN0370
00267      39*           L=L+15
00270      40*           J=J+15
00271      41*           IF(J.GE.LNCAT)J=LNCAT                                    PRIN0400
00273      42*           GO TO 60                                                 PRIN0410
00274      43*        80 CONTINUE                                                 PRIN0420
00275      44*           IF (KKT.EQ.-1) GO TO 90                                  PRIN0430
00277      45*           IF(MOD(KKT, MAP).NE.0)RETURN                             PRIN0440
00301      46*        90 CONTINUE                                                 PRIN0450
00302      47*           DO 100 I=1,LNCAT                                         PRIN0460
```

```
00305           DO 100 K=1,NOFLD                                    PRIN0470
00310       100 NBLK(K,I)=0                                         PRIN0480
00313           IRC=IRD                                             PRIN0490
00314           ICCT=NOPTS                                          PRIN0500
00315           IF(IRD.EQ.0)ICCT=KPTS                               PRIN0510
00317           IF (IRD.EQ.0) GO TO 110                             PRIN0520
00321           CALL NTRAN(7,2,ICCT,IPLACE,ISTAT,22)                PRIN0530
00322           IREC=1                                              PRIN0540
00323       110 JPTS=0                                              PRIN0550
00324           IRC=IRC-1                                           PRIN0560
00325           CALL SETMRG(66,0,66)                                PRIN0570
00326           DO 200 IFLD=1,NOFLD                                 PRIN0580
00331           LINES=BLK(IFLD,1)                                   PRIN0590
00332           PTS=BLK(IFLD,2)                                     PRIN0600
00333           IB=BLOCK(IFLD,4)                                    PRIN0610
00334           IE=BLOCK(IFLD,5)                                    PRIN0620
00335           INC=BLOCK(IFLD,6)                                   PRIN0630
00336           J=0                                                 PRIN0640
00337       120 DO 130 I=IB,IE,INC                                  PRIN0650
00342           J=J+1                                               PRIN0660
00343           COL(1,J)=I/100                                      PRIN0670
00344           COL(2,J)=MOD(I,100)/10                              PRIN0680
00345           COL(3,J)=MOD(I,10)                                  PRIN0690
00346           IF (J.EQ.110) GO TO 140                             PRIN0700
00350       130 CONTINUE                                            PRIN0710
00352       140 LPTS=J                                              PRIN0720
00353           IF (IFLD.NE.1) WRITE (6,360)                        PRIN0730
00356           WRITE (6,220)                                       PRIN0740
00360           WRITE (6,HEAD)                                      PRIN0750
00362           J=PTS*LINES                                         PRIN0760
00363           WRITE (6,330)FLDNAM(IFLD),J,(BLOCK(IFLD,I),I=1,6)   PRIN0770
00373           DO 150 I=1,3                                        PRIN0780
00376       150 WRITE (6,210)(COL(I,J),J=1,LPTS)                    PRIN0790
00405           WRITE (6,220)                                       PRIN0800
00407           LINE=BLOCK(IFLD,1)                                  PRIN0810
00410           DO 180 I=1,LINES                                    PRIN0820
00413           DO 170 J=1,PTS                                      PRIN0830
00416           JPTS=JPTS+1                                         PRIN0840
00417           IF (JPTS.LE.ICCT) GO TO 160                         PRIN0850
00421           IF(IRC.EQ.1)ICCT=KPTS                               PRIN0860
00423           CALL NTRAN(7,2,ICCT,IPLACE,ISTAT,22)                PRIN0870
00424           IRC=IRC-1                                           PRIN0880
00425           JPTS=1                                              PRIN0890
00426       160 CONTINUE                                            PRIN0900
00427           K=IPLACE(JPTS)                                      PRIN0910
00430           NBLK(IFLD,K)=NBLK(IFLD,K)+1                         PRIN0920
00431           IF (J.GT.110) GO TO 170                             PRIN0930
00433           OUT(J)=SYMBLS(K)                                    PRIN0940
00434       170 CONTINUE                                            PRIN0950
00436           WRITE (6,230)LINE,(OUT(J),J=1,LPTS)                 PRIN0960
00445       180 LINE=BLOCK(IFLD,3)+LINE                             PRIN0970
00447           WRITE (6,370)                                       PRIN0980
00451           DO 190 I=1,LNCAT                                    PRIN0990
00454       190 WRITE (6,380)I,SYMBLS(I),NBLK(IFLD,I)               PRIN1000
00462       200 CONTINUE                                            PRIN1010
00464           CALL SETMRG(66,4,62)                                PRIN1020
00465           RETURN                                              PRIN1030
00466       210 FORMAT(9X,110I1)                                    PRIN1040
```

```
00467     106*        220 FORMAT(/)                                                    PRIN1050
00470     107*        230 FORMAT(2X,I5,2X,110A1)                                       PRIN1060
00471     108*        240 FORMAT(//' INTERMEDIATE PRINTOUT FOR ITERATION',I5//)        PRIN1070
00472     109*        250 FORMAT(//' TOTAL NUMBER OF CLUSTERS =',I3)                   PRIN1080
00473     110*        260 FORMAT(/' TOTAL NUMBER OF POINTS =',I7)                      PRIN1090
00474     111*        270 FORMAT(//' CLUSTER    SYMBOL    POINTS IN CLUSTER')          PRIN1100
00475     112*        280 FORMAT(4X,I2,9X,A1,10X,I7)                                   PRIN1110
00476     113*        290 FORMAT(///15X,'MEANS'/)                                      PRIN1120
00477     114*        300 FORMAT(/2X,'CLUSTER',5X,12(A1,'CH(',I2,')',1X))              PRIN1130
00500     115*        310 FORMAT(5X,I2,7X,12(F7.2,1X))                                 PRIN1140
00501     116*        320 FORMAT(///10X,' STANDARD DEVIATIONS'/)                       PRIN1150
00502     117*        330 FORMAT(//2X,A6,//' TOTAL NUMBER OF POINTS IN THIS FIELD',I7/ PRIN1160
00502     118*           * ' LINE',I5,' THROUGH'                                       PRIN1170
00502     119*           * ,I5,' EVERY',I3,' LINE(S)   SAMPLE',I4,' THROUGH',I4,' EVERY',  PRIN1180
00502     120*           * I3,' SAMPLE(S)'///)                                         PRIN1190
00503     121*        340 FORMAT(//15X,'DISTANCES BETWEEN CLUSTERS'//1X,'CLUSTER',15I8)
00504     122*        350 FORMAT(3X,I2,5X,15F8.2)
00505     123*        360 FORMAT(1H1)                                                  PRIN1220
00506     124*        370 FORMAT(//2X,'POINTS PER CLUSTER IN THIS FIELD'/3X,'CLUSTER',   PRIN1230
00506     125*           *  5X,'SYMBOL',5X,'POINTS'/)                                  PRIN1240
00507     126*        380 FORMAT(6X,I2,10X,A1,7X,I5)                                   PRIN1250
00510     127*            END                                                          PRIN1260

     END OF COMPILATION:          NO  DIAGNOSTICS.
   PRINT              SYMBOLIC                    29 NOV 73  13:41:07   0  01600002    14    127  (DELETED)
   PRINT    CODE      RELOCATABLE                 29 NOV 73  13:41:07   1  01603364    24      1  (DELETED)
                                                                        0  01603414    14     70
```

```
@ FOR,* SETUP,SETUP                                                                                    03 DEC 73          12:23: 7.586
UNIVAC 1108 FORTRAN V EXEC II LEVEL 25A  -(EXEC8 LEVEL E12010010A)
THIS COMPILATION WAS DONE ON 03 DEC 73 AT 12:23:07

    SUBROUTINE SETUP      ENTRY POINT 001477

    STORAGE USED: CODE(1) 001511; DATA(0) 000701; BLANK COMMON(2) 000000

     COMMON BLOCKS:

     0003     PASS    026574

    EXTERNAL REFERENCES (BLOCK, NAME)

     0004     FLTNUM
     0005     NXTCHR
     0006     NUMBER
     0007     EXIT
     0010     DRMAVL
     0011     DRMASG
     0012     NTRAN
     0013     NWDUS
     0014     NIO2$
     0015     NRDUS
     0016     NIO1$
     0017     NERR2$
     0020     NERR3$

    STORAGE ASSIGNMENT  (BLOCK, TYPE, RELATIVE LOCATION, NAME)

     0001     000017 10L        0001     000312 100L       0001     000322 110L       0001     000334 120L       0001     000360 130L
     0001     000373 140L       0001     000406 150L       0001     000421 160L       0001     000027 167G       0001     000434 170L
     0001     000042 176G       0001     000435 180L       0001     000449 190L       0001     000464 200L       0001     000051 209G
     0001     000510 210L       0001     000123 219G       0001     000534 220L       0001     000562 225L       0001     000154 227G
     0001     000605 230L       0001     000631 235L       0001     000643 240L       0001     000656 250L       0001     000666 260L
     0001     000722 265L       0001     000776 280L       0001     001031 290L       0001     000130 30L        0001     001070 300L
     0001     001107 310L       0001     000366 311G       0001     000401 320G       0001     001126 320L       0001     000414 327G
     0001     001146 330L       0001     000427 336G       0001     001165 340L       0001     001204 350L       0001     001210 360L
     0001     001225 370L       0001     001275 400L       0001     001312 420L       0001     001345 435L       0001     001357 440L
     0001     000711 441G       0001     000712 443G       0001     001370 445L       0001     000746 455G       0001     001416 460L
     0001     000756 463G       0001     000765 470G       0001     001436 470L       0001     001445 475L       0000     000301 480F
     0000     000303 490F       0001     000160 50L        0000     000313 500F       0001     001005 500G       0000     000315 510F
     0001     001045 517G       0000     000317 520F       0000     000321 530F       0000     000330 540F       0000     000337 550F
     0000     000342 560F       0000     000351 570F       0000     000366 580F       0000     000400 590F       0001     000204 60L
     0000     000413 600F       000C     000430 610F       0001     001242 613G       0000     000442 620F       0001     001252 621G
     0000     000454 630F       0001     001271 632G       0000     000460 640F       0001     001307 642G       0000     000470 650F
     0001     001317 651G       0000     000474 660F       0000     000502 670F       0000     000546 680F       0000     000554 690F
     0001     000232 70L        0000     000613 700F       0000     000624 710F       0001     000256 80L        0001     000302 90L
     0000 L   000266 ASGDRM     0000 I   000163 BLANK      0003 I   007305 BLK        0003 I   006155 BLOCK      0000 I   000026 CARD
     0003 R   026507 CHNTHS     0003     007706 CLD        0000 I   000162 CODE       0000 I   000164 COL        0003 I   000037 COMNT
     0003 I   007677 DAS        0003 I   000016 DATE       0003 R   000116 DLMIN      0003 I   000053 FETVEC     0003 I   006011 FLDNAM
     0004 I   000000 FLTNUM     0003 I   007700 FORMAT     0003 I   000000 HEAD       0003 I   000002 HED1       0003 I   000023 HED2
     0000 I   000267 I          0003     026510 ICHAIN     0003 I   026506 ICHN       0000 I   000271 ICNT       0000     000655 INJPS
     0000 I   000000 INVEC      0000 I   000265 IPASS      0000 I   000279 IPTS       0003     007703 IRD        0003 I   000111 ISTOP
```

```
0000 I 000275 ITOT       0000 I 000270 J          0000 I 000273 K          0003   007704 KPTS       0003 I 000114 KRN
0003 I 026573 KUNIT      0003 I 000112 LNCAT      0000 I 000272 M          0003 I 007701 MAP        0003 I 026505 MAXCLS
0003 R 000120 MEANS      0000 I 000174 MTABLE     0003   014612 N          0003   014674 NBLK       0003 I 000113 NMIN
0003 I 000052 NOFEAT     0003 I 006010 NOFLD      0003   007705 NOPTS      0000 I 000300 NSTAT      0000 I 000165 NTABLE
0006 I 000000 NUMBER     0000 I 000124 NUMVEC     0005 I 000000 NXTCHR     0000 I 000277 N1762      0000 I 000276 N932
0003 I 026504 PUNCH      0003 R 000117 SEP        0003 I 000203 SMBLS      0003 I 007702 SPTRIG     0003   003054 STDEV
0003 R 000115 STDMAX     0003 I 007615 SYMBLS     0003 I 026572 VARSIZ


00101      1*         SUBROUTINE SETUP                                                SETU0000
00101      2*      C*********************************************************************SETU0010
00101      3*      C                                                                  SETU0020
00101      4*      C     THE PURPOSE OF SUBROUTINE SETUP IS TO READ AND ANALYZE ALL CARD  SETU0030
00101      5*      C     INPUT TO THE PROGRAM, INITIALIZE DEFAULT OPTIONS,AND ASSIGN DRUM SETU0040
00101      6*      C     STORAGE IF NEEDED                                            SETU0050
00101      7*      C                                                                  SETU0060
00101      8*      C*********************************************************************SETU0070
00101      9*      C                                                                  SETU0080
00103     10*         INCLUDE COMMON,LIST                                             SETU0090
00104     10*         PARAMETER MAXPOP=50,MAXFET=30
00105     10*         PARAMETER MAXDIM=25000
00106     10*         PARAMETER MAXFLD=100
00107     10*         COMMON/PASS/HEAD(42),NOFEAT,FETVEC(MAXFET),ISTOP,LNCAT,NMIN,KRN,
00107     10*        * STDMAX,DLMIN,SEP,MEANS(MAXFET,MAXPOP),STDEV(MAXFET,MAXPOP),
00107     10*        * NOFLD,FLDNAM(MAXFLD),BLOCK(MAXFLD,6),BLK(MAXFLD,2),
00107     10*        * SYMBLS(MAXPOP),DAS,FORMAT,MAP,
00107     10*        * SPTRIG,IRD,KPTS,NOPTS,CLD(MAXPOP,MAXPOP)
00107     10*        *, N(MAXPOP),NBLK(MAXFLD,MAXPOP)
00107     10*        * ,PUNCH,MAXCLS,ICHN,CHNTHS,ICHAIN(MAXPOP),VARSIZ
00107     10*        *,KUNIT
00110     10*         INTEGER VARSIZ
00111     10*         INTEGER PUNCH
00112     10*         REAL MEANS
00113     10*         INTEGER SPTRIG,BLOCK,BLK,SYMBLS,HEAD,FETVEC,FLDNAM,DAS
00114     10*         INTEGER FORMAT
00115     10*         END
00116     11*         DIMENSION INVEC(22),CARD(62),NUMVEC(30),DATE(2)
00117     12*         DIMENSION HED1(10),HED2(10),COMNT(10)                           SETU0110
00120     13*         EQUIVALENCE (HED1(1),HEAD(3)),     (DATE(1),HEAD(15)),          SETU0120
00120     14*        *            (HED2(1),HEAD(20)),       (COMNT(1),HEAD(32))       SETU0130
00121     15*         DATA INVEC/'FEATUR','ISTOP','LNCAT','NMIN','KRN','STDMAX','DLMIN',SETU0140
00121     16*        *'SEP','FORMAT','HED1','HED2','DATE','*END*','SEND*','COMMEN',    SETU0150
00121     17*        * 'SYMBOL','DASUNI','MAP','PUNCH','MAXCLS','CHAIN','ERIPS'/
00123     18*         INTEGER CODE,CARD,DATE,HED1,HED2,BLANK,COL,COMNT,FLTNUM         SETU0170
00124     19*         DIMENSION NTABLE(7),MTABLE(7)                                   SETU0180
00125     20*         INTEGER SMBLS                                                   SETU0190
00126     21*         DIMENSION SMBLS(50)                                             SETU0200
00126     22*      C*                                                                 SETU0210
00126     23*      C*     DEFAULT VALUES FOR INPUT PARAMETERS                         SETU0220
00126     24*      C*                                                                 SETU0230
00127     25*         DATA PUNCH/0/                                                   SETU0240
00131     26*         DATA DLMIN/3.2/,FORMAT/2/,STDMAX/4.5/,MAP/20/
00136     27*         DATA ISTOP/10/,NMIN/30/,KRN/1/                                  SETU0260
00142     28*         DATA MAXCLS/MAXPOP/                                             SETU0270
00144     29*         DATA SMBLS/'1','2','3','4','5','6','7','8','9','A','B','C','D','E'SETU0280
00144     30*        *,'F','G','H','I','J','K','L','M','N','O','P','Q','R','S','T','U', SETU0290
```

```
00144        31*          *'V','W','X','Y','Z','8','=','&','/','-','.','+','$','@','*','6','?SETU0300
00144        32*          *','IH',',',' '/                                                    SETU0310
00146        33*           DATA BLANK/ ' '/                                                   SETU0320
00150        34*           DATA IPASS/0/,ICHN/0/
00153        35*           LOGICAL ASGORM
00154        36*           DATA ASGORM/.FALSE./
00156        37*           IPASS=IPASS+1
00157        38*           NOFLD=1                                                            SETU0330
00160        39*           WRITE (6,HEAD)                                                     SETU0340
00162        40*           WRITE (6,630)                                                      SETU0350
00164        41*        10 READ(5,480,END=475)CODE,CARD
00173        42*           WRITE (6,550)CODE,CARD                                             SETU0370
00202        43*           COL=0                                                              SETU0380
00203        44*           DO 20   I=1,22
00206        45*        20 IF(CODE .EQ. INVEC(I)) GO TO (30,50,60,70,80,90,100,110,120,130,
00206        46*          *140,150,240,360,160,170,200,210,220,230,235,225),I
00211        47*           WRITE (6,490)CODE,CARD                                             SETU0420
00220        48*           GO TO 10                                                           SETU0430
00220        49*     C*                                                                       SETU0440
00220        50*     C*    FEATURE CARD                                                       SETU0450
00220        51*     C*                                                                       SETU0460
00221        52*        30 J = NXTCHR(CARD,COL)                                               SETU0470
00222        53*           IF (J.EQ.BLANK) GO TO 10                                           SETU0480
00224        54*           COL=COL-1                                                          SETU0490
00225        55*           NOFEAT=NUMBER(CARD,COL,NUMVEC)                                     SETU0500
00226        56*           DO 40 I=1,NOFEAT                                                   SETU0510
00231        57*        40 FETVEC(I)=NUMVEC(I)                                                SETU0520
00233        58*           GO TO 10                                                           SETU0530
00233        59*     C*                                                                       SETU0540
00233        60*     C*    ISTOP CARD (MAXIMUM NUMBER OF ITERATIONS)                          SETU0550
00233        61*     C*                                                                       SETU0560
00234        62*        50 J = NXTCHR(CARD,COL)                                               SETU0570
00235        63*           IF (J.EQ.BLANK) GO TO 10                                           SETU0580
00237        64*           COL=COL-1                                                          SETU0590
00240        65*           J = NUMBER(CARD,COL,NUMVEC)                                        SETU0600
00241        66*           ISTOP = NUMVEC(1)                                                  SETU0610
00242        67*           GO TO 10                                                           SETU0620
00242        68*     C*                                                                       SETU0630
00242        69*     C*    LNCAT CARD (NUMBER OF INITIAL CLUSTERS OR CATEGORIES)              SETU0640
00242        70*     C*                                                                       SETU0650
00243        71*        60 J = NXTCHR(CARD,COL)                                               SETU0660
00244        72*           IF (J.EQ.BLANK) GO TO 10                                           SETU0670
00246        73*           COL = COL-1                                                        SETU0680
00247        74*           J = NUMBER(CARD,COL,NUMVEC)                                        SETU0690
00250        75*           LNCAT = NUMVEC(1)                                                  SETU0700
00251        76*           IPASS=1
00252        77*           GO TO 10                                                           SETU0710
00252        78*     C*                                                                       SETU0720
00252        79*     C*    NMIN CARD (MINIMUM NUMBER OF POINTS PER CLUSTER)                   SETU0730
00252        80*     C*                                                                       SETU0740
00253        81*        70 J = NXTCHR(CARD,COL)                                               SETU0750
00254        82*           IF (J.EQ.BLANK) GO TO 10                                           SETU0760
00256        83*           COL = COL-1                                                        SETU0770
00257        84*           J = NUMBER(CARD,COL,NUMVEC)                                        SETU0780
00260        85*           NMIN = NUMVEC(1)                                                   SETU0790
00261        86*           GO TO 10                                                           SETU0800
00261        87*     C*                                                                       SETU0810
00261        88*     C*    KRN CARD (NUMBER OF ITERATIONS PER FULL OUTPUT)                    SETU0820
```

ORIGINAL PAGE IS
OF POOR QUALITY

```
00261      C*                                                          SETU0830
00262         80 J = NXTCHR(CARD,COL)                                  SETU0840
00263            IF (J.EQ.BLANK) GO TO 10                              SETU0850
00265            COL = COL-1                                           SETU0860
00266            J = NUMBER(CARD,COL,NUMVEC)                           SETU0870
00267            KRH = NUMVEC(1)                                       SETU0880
00270            GO TO 10                                              SETU0890
00270      C*                                                          SETU0900
00270      C*    STDMAX CARD (MAXIMUM STANDARD DEVIATION PER CLUSTER)  SETU0910
00270      C*                                                          SETU0920
00271         90 J = FLTNUM(CARD,COL,STDMAX,1)                         SETU0930
00272            GO TO 10                                              SETU0940
00272      C*                                                          SETU0950
00272      C*    DLMIN CARD (MINIMUM DISTANCE BETWEEN CLUSTER MEANS)   SETU0960
00272      C*                                                          SETU0970
00273        100 J = FLTNUM(CARD,COL,DLMIN,1)                          SETU0980
00274            GO TO 10                                              SETU0990
00274      C*                                                          SETU1000
00274      C*    SEP CARD  (DISTANCE FOR SPLITTING)                    SETU1010
00274      C*                                                          SETU1020
00275        110 J = FLTNUM(CARD,COL,SEP,1)                            SETU1030
00276            SPTRIG=1                                              SETU1040
00277            GO TO 10                                              SETU1050
00277      C*                                                          SETU1060
00277      C*    FORMAT CARD                                           SETU1070
00277      C*                                                          SETU1080
00300        120 J=NXTCHR(CARD,COL)                                    SETU1090
00301            IF (J.EQ.BLANK) GO TO 10                              SETU1100
00303            COL=COL-1                                             SETU1110
00304            J=NUMBER(CARD,COL,NUMVEC)                             SETU1120
00305            FORMAT=NUMVEC(1)                                      SETU1130
00306            GO TO 10                                              SETU1140
00306      C*                                                          SETU1150
00306      C*    HED1 CARD                                             SETU1160
00306      C*                                                          SETU1170
00307        130 READ (30,500)HED1                                     SETU1180
00315            GO TO 10                                              SETU1190
00315      C*                                                          SETU1200
00315      C*    HED2 CARD                                             SETU1210
00315      C*                                                          SETU1220
00316        140 READ (30,500)HED2                                     SETU1230
00324            GO TO 10                                              SETU1240
00324      C*                                                          SETU1250
00324      C*    DATE CARD                                             SETU1260
00324      C*                                                          SETU1270
00325        150 READ (30,500)DATE                                     SETU1280
00333            GO TO 10                                              SETU1290
00333      C*                                                          SETU1300
00333      C*    COMMENT CARD                                          SETU1310
00333      C*                                                          SETU1320
00334        160 READ (30,500)COMNT                                    SETU1330
00342            GO TO 10                                              SETU1340
00342      C*                                                          SETU1350
00342      C*    SYMBOLS CARD                                          SETU1360
00342      C*                                                          SETU1370
00343        170 ICNT=0                                                SETU1380
00344        180 ICNT=ICNT + 1                                         SETU1390
00345            IF (ICNT.GT.MAXPOP) GO TO 10                          SETU1400
```

```
00347        190 M=NXTCHR(CARD,COL)                                               SETU1410
00350            IF (M.EQ.BLANK) GO TO 10                                         SETU1420
00352            IF (M.EQ.',') GO TO 190                                          SETU1430
00354            SYMBLS(ICNT)=M                                                   SETU1440
00355            GO TO 180                                                        SETU1450
00355      C*                                                                     SETU1460
00355      C*    DASUNIT CARD (UNIT ON WHICH DAS TAPE IS TO BE CREATED)           SETU1470
00355      C*                                                                     SETU1480
00356        200 J=NXTCHR(CARD,COL)                                               SETU1490
00357            IF (J.EQ.BLANK) GO TO 10                                         SETU1500
00361            COL=COL-1                                                        SETU1510
00362            J=NUMBER(CARD,COL,NUMVEC)                                        SETU1520
00363            DAS=NUMVEC(1)                                                    SETU1530
00364            GO TO 10                                                         SETU1540
00364      C*                                                                     SETU1550
00364      C*    MAP CARD  (NUMBER OF ITERATIONS TO OUTPUT MAP)                   SETU1560
00364      C*                                                                     SETU1570
00365        210 J=NXTCHR(CARD,COL)                                               SETU1580
00366            IF (J.EQ.BLANK) GO TO 10                                         SETU1590
00370            COL=COL-1                                                        SETU1600
00371            J=NUMBER(CARD,COL,NUMVEC)                                        SETU1610
00372            MAP=NUMVEC(1)                                                    SETU1620
00373            GO TO 10                                                         SETU1630
00373      C*                                                                     SETU1640
00373      C*    PUNCH CARD (PUNCH SJATISTICS ON CARDS)                           SETU1650
00373      C*                                                                     SETU1660
00374        220 PUNCH=3                                                          SETU1670
00375            J=NXTCHR(CARD,COL)                                               SETU1680
00376            IF (J.EQ.BLANK) GO TO 10                                         SETU1690
00400            COL=COL-1                                                        SETU1700
00401            J=NUMBER(CARD,COL,NUMVEC)                                        SETU1710
00402            PUNCH=NUMVEC(1)                                                  SETU1720
00403            GO TO 10                                                         SETU1730
00403      C
00403      C     ERIPS CARD ( UNIT TO USE WHEN CREATING A TAPE )
00403      C
00404        225 J=NXTCHR(CARD,COL)
00405            IF (J .EQ. BLANK)  GO TO 10
00407            COL = COL - 1
00410            J = NUMBER(CARD,COL,NUMVEC)
00411            KUNIT = NUMVEC(1)
00411      C*                                                                     SETU1740
00411      C*    MAXCLS CARD  (MAXIMUM NUMBER OF CLUSTERS)                        SETU1750
00411      C*                                                                     SETU1760
00412        230 J=NXTCHR(CARD,COL)                                               SETU1770
00413            IF (J.EQ.3LANK) GO TO 10                                         SETU1780
00415            COL=COL-1                                                        SETU1790
00416            J=NUMBER(CARD,COL,NUMVEC)                                        SETU1800
00417            MAXCLS = NUMVEC(1)                                               SETU1810
00420            GO TO 10                                                         SETU1820
00420      C*
00420      C*    CHAIN CARD (CHAIN CLUSTERS WHICH ARE DLMIN UNITS APART)
00421        235 ICHN=1
00422            J=FLTNUM(CARD,COL,CHNTHS,1)
00423            GO TO 10
00423      C*                                                                     SETU1830
00423      C*    *END* CARD (IF THE NUMBER OF ORIGINAL CLUSTERS IS GREATER THAN ONESETU1840
00423      C*                  READ INITIAL MEANS)                                SETU1850
```

```
00423      C*                                                              SETU1860
00423      C*                                                              SETU1870
00424        240 IF (NOFEAT.GT.0) GO TO 250                                SETU1880
00426            WRITE (6,560)                                             SETU1890
00430            CALL EXIT                                                 SETU1900
00431        250 IF (LNCAT.GE.2) GO TO 260                                 SETU1910
00433            LNCAT=1                                                   SETU1920
00434            GO TO 280                                                 SETU1930
00435        260 IF(IPASS.GT.1)GO TO 265
00437            READ(5,510)((MEANS(J,K),J=1,NOFEAT),K=1,LNCAT)
00450        265 CONTINUE
00451            WRITE (6,660)                                             SETU1950
00453            WRITE (6,640)(BLANK,FETVEC(I),I=1,NOFEAT)                 SETU1960
00462            DO 270 J=1,LNCAT                                          SETU1970
00465        270 WRITE (6,650)J,(MEANS(I,J),I=1,NOFEAT)                    SETU1980
00465      C*                                                              SETU1990
00465      C*    READ IN FIELD IDENTIFIERS                                 SETU2000
00465      C*                                                              SETU2010
00475        280 READ (5,480)CODE,CARD                                     SETU2020
00504            IF (CODE.EQ.'SEND*') GO TO 360                            SETU2030
00506            IF (NOFLD.LE.MAXFLD) GO TO 290                            SETU2040
00510            WRITE (6,540)                                             SETU2050
00512            CALL EXIT                                                 SETU2060
00513        290 CONTINUE                                                  SETU2070
00514            READ (30,520)FLDNAM(NOFLD),(BLOCK(NOFLD,J),J=1,6)         SETU2080
00514      C*                                                              SETU2090
00514      C*    ERROR CHECKING FOR INPUT FIELDS                           SETU2100
00514      C*                                                              SETU2110
00523            IF ((BLOCK(NOFLD,2)-BLOCK(NOFLD,1)).GE.0) GO TO 300       SETU2120
00525            WRITE (6,570)NOFLD,BLOCK(NOFLD,1),BLOCK(NOFLD,2)          SETU2130
00532            CALL EXIT                                                 SETU2140
00533        300 IF (BLOCK(NOFLD,1).GE.1) GO TO 310                        SETU2150
00535            WRITE (6,580)NOFLD,BLOCK(NOFLD,1)                         SETU2160
00541            CALL EXIT                                                 SETU2170
00542        310 IF (BLOCK(NOFLD,3).GE.1) GO TO 320                        SETU2180
00544            WRITE (6,590)NOFLD,BLOCK(NOFLD,3)                         SETU2190
00550            CALL EXIT                                                 SETU2200
00551        320 IF ((BLOCK(NOFLD,5)-BLOCK(NOFLD,4)).GE.0) GO TO 330       SETU2210
00553            WRITE (6,600)NOFLD,BLOCK(NOFLD,4),BLOCK(NOFLD,5)          SETU2220
00560            CALL EXIT                                                 SETU2230
00561        330 IF (BLOCK(NOFLD,4).GE.1) GO TO 340                        SETU2240
00563            WRITE (6,610)NOFLD,BLOCK(NOFLD,4)                         SETU2250
00567            CALL EXIT                                                 SETU2260
00570        340 IF (BLOCK(NOFLD,6).GE.1) GO TO 350                        SETU2270
00572            WRITE (6,620)NOFLD,BLOCK(NOFLD,6)                         SETU2280
00576            CALL EXIT                                                 SETU2290
00577        350 CONTINUE                                                  SETU2300
00600            NOFLD = NOFLD+1                                           SETU2310
00601            GO TO 280                                                 SETU2320
00601      C*                                                              SETU2330
00601      C*    SEND CARD                                                 SETU2340
00601      C*                                                              SETU2350
00602        360 NOFLD = NOFLD-1                                           SETU2360
00603            IF (NOFLD.GT.0) GO TO 370                                 SETU2370
00605            WRITE (6,530)                                             SETU2380
00607            CALL EXIT                                                 SETU2390
00610        370 WRITE (6,670)                                             SETU2400
00612     c      DO 380 I=1,NOFLD                                          SETU2410
```

```
00615                WRITE (6,680)I,FLDNAM(I),(BLOCK(I,J),J=1,6)                SETU2420
00625            380 CONTINUE                                                   SETU2430
00625      C*                                                                   SETU2440
00625      C*                                                                   SETU2441
00625      C*        ARE DEFAULT SYMBOLS TO BE USED?                            SETU2442
00625      C*                                                                   SETU2443
00627                IF(ICNT.GT.M)GO TO 400                                     SETU2444
00631                DO 390 I=1,MAXCLS                                          SETU2445
00634            390 SYMBLS(I)=SMBLS(I)                                         SETU2450
00636                GO TO 420                                                  SETU2460
00637            400 IF (ICNT.GE.MAXCLS) GO TO 420                              SETU2470
00641                DO 410 I=ICNT,MAXCLS                                       SETU2480
00644            410 SYMBLS(I)=SMBLS(I)                                         SETU2490
00646            420 CONTINUE                                                   SETU2500
00646      C*                                                                   SETU2510
00646      C*        ASSIGN DRUM STORAGE IF NEEDED                              SETU2520
00646      C*                                                                   SETU2530
00647                IPTS=0                                                     SETU2540
00650                DO 430 I=1,NOFLD                                           SETU2550
00653                IPTS=IPTS+((BLOCK(I,2)-BLOCK(I,1))/BLOCK(I,3) + 1) *       SETU2560
00653          *        ((BLOCK(I,5)-BLOCK(I,4))/BLOCK(I,6) + 1)                SETU2570
00654            430 CONTINUE                                                   SETU2580
00656                ITOT = IPTS*NOFEAT                                         SETU2590
00657                IF(ASGDRM)GO TO 435
00661                CALL DRMAVL(N432,N1782)                                    SETU2600
00662            435 CONTINUE
00663                IF (ITOT.GT.N432) GO TO 440                                SETU2620
00665                IF(IPTS.LE.N1782)GO TO 445
00667            440 WRITE (6,690)N432,N1782                                    SETU2660
00673                CALL EXIT                                                  SETU2670
00674            445 IF(ASGDRM)GO TO 470
00676                ASGDRM=.TRUE.
00677                ITOT=N432
00700            450 CALL DRMASG(4,ITOT,NTABLE,NSTAT)                           SETU2680
00701                IF (NSTAT.EQ.0) GO TO 460                                  SETU2690
00703                WRITE (6,700)NSTAT                                         SETU2700
00706                CALL EXIT                                                  SETU2710
00707            460 CALL DRMASG(7,N1782,MTABLE,NSTAT)
00710                IF (NSTAT.EQ.0) GO TO 470                                  SETU2730
00712                WRITE (6,710)NSTAT                                         SETU2740
00715                CALL EXIT                                                  SETU2750
00716            470 CONTINUE                                                   SETU2760
00716      C*                                                                   SETU2770
00717                CALL NTRAN(4,10)
00717      C*        RETURN TO PROCESS THIS SET OF DATA
00720                RETURN                                                     SETU2780
00720      C*                                                                   SETU2790
00720      C*        FINISHED. REWIND OUTPUT DAS TAPE AND EXIT
00721            475 IF(DAS.GT.0)CALL NTRAN(DAS,11,22)
00723                CALL EXIT
00724            480 FORMAT(A6,4X,62A1)                                         SETU2800
00725            490 FORMAT(' INVALID INPUT CARD--IGNORED'/T5,A6,4X,62A1)       SETU2810
00726            500 FORMAT(10X,10A6)                                           SETU2820
00727            510 FORMAT(8F10.2)                                             SETU2830
00730            520 FORMAT(A6,14X,6I5)                                         SETU2840
00731            530 FORMAT(' AT LEAST ONE FIELD MUST BE INPUT')                SETU2850
00732            540 FORMAT(' TOO MANY FIELDS ARE BEING INPUT')                 SETU2860
00733            550 FORMAT(5X,A6,4X,62A1)                                      SETU2870
```

ORIGINAL PAGE IS
OF POOR QUALITY

```
00734      560 FORMAT(' AT LEAST ONE FEATURE MUST BE INPUT')                     SETU2880
00735      570 FORMAT(' IN FIELD',I4,' INITIAL LINE',I5,' IS GREATER THAN LAST LISETU2890
00735         *NE',I5)                                                           SETU2900
00736      580 FORMAT(' INITIAL LINE IN FIELD',I4,' DEFINED LESS THAN 1', I5)    SETU2910
00737      590 FORMAT(' LINE INCREMENT IN FIELD',I4,' DEFINED LESS THAN 1',I5)   SETU2920
00740      600 FORMAT(' IN FIELD',I4,' INITIAL SAMPLE',I5,' IS GREATER THAN LAST SETU2930
00740         *SAMPLE',I5)                                                       SETU2940
00741      610 FORMAT(' INITIAL SAMPLE IN FIELD',I4,' IS LESS THAN 1',I5)        SETU2950
00742      620 FORMAT(' SAMPLE INCREMENT IN FIELD',I4,' IS LESS THAN 1',I5)      SETU2960
00743      630 FORMAT(//' INPUT SUMMARY'//)                                      SETU2970
00744      640 FORMAT(2X,'CLUSTER',2X,12(A1,'CH('I2,')',1X))                     SETU2980
00745      650 FORMAT(5X,I2,5X,12(F6.2,2X))                                      SETU2990
00746      660 FORMAT(/15X,'INITIAL CLUSTER MEANS'/)                             SETU3000
00747      670 FORMAT(//30X,'FIELDS TO BE CLUSTERED'//10X,'FIELD',5X,'FIRST',5X, SETU3010
00747         * 'LAST',6X,'LINE',6X,'FIRST',5X,'LAST',6X,'SAMPLE'/10X,'NAME',6X, SETU3020
00747         * 'LINE',6X,'LINE',6X,'INC.',6X,'SAMPLE',4X,'SAMPLE',4X,'INC.'/)   SETU3030
00750      680 FORMAT(3X,I3,'.',3X,A6,4X,6(I5,5X))                               SETU3040
00751      690 FORMAT(' USER HAS REQUESTED TOO MANY DATA POINTS--STORAGE NOT AVAISETU3050
00751         *LABLE'//' AVAILABLE STORAGE ON FH432 DRUMS IS',I7,' WORDS'/       SETU3060
00751         * ' AVAILABLE STORAGE ON FH1782 DRUM IS',I7,' WORDS'/)             SETU3070
00752      700 FORMAT(' FILE ASSIGNMENT NOT MADE FOR UNIT 4---NSTAT=',I7)        SETU3080
00753      710 FORMAT(' FILE ASSIGNMENT NOT MADE FOR UNIT 7---NSTAT=',I7)        SETU3090
00754          END                                                               SETU3100


      END OF COMPILATION:          NO  DIAGNOSTICS.
   SETUP             SYMBOLIC                                29 NOV 73  13:41:11   0  01605340    14   343  (DELETED)
   SETUP    CODE     RELOCATABLE                             29 NOV 73  13:41:11   1  01616642    36     1  (DELETED)
                                                                                   0  01616706    14   118
```

```
@ FOR,* TAPERD,TAPERD                                                                                03 DEC 73          12:23:11.499
UNIVAC 1108 FORTRAN V EXEC II LEVEL 25A  -(EXEC8 LEVEL E12010010A)
THIS COMPILATION WAS DONE ON 03 DEC 73 AT 12:23:11

    SUBROUTINE TAPERD     ENTRY POINT 001400

    STORAGE USED: CODE(1) 001635; DATA(0) 000723; BLANK COMMON(2) 000000

     COMMON BLOCKS:

     0003     PASS    026574

    EXTERNAL REFERENCES (BLOCK, NAME)

     0004     NTRAN
     0005     CMERR
     0006     UNPAK1
     0007     UNPACK
     0010     UNPCK
     0011     NWDUS
     0012     NI02$
     0013     NI01$
     0014     NEXP1$
     0015     NERR3$

    STORAGE ASSIGNMENT  (BLOCK, TYPE, RELATIVE LOCATION, NAME)

     0001     000273 100L       0001     000343 120L       0001     000407 130L       0001     000455 140L       0001     000475 150L
     0001     000062 156G       0001     000527 160L       0001     000677 170L       0001     000123 176G       0001     000707 180L
     0001     000051 20L        0001     000717 200L       0001     000727 220L       0001     000750 230L       0001     001017 240L
     0001     000300 251G       0001     001116 260L       0001     001120 270L       0001     001140 280L       0001     001205 290L
     0001     001224 300L       0001     001512 300L       0001     001306 320L       0001     000463 326G       0000     000234 330F
     0000     000244 340F       0000     000254 350F       0000     000263 360F       0000     000273 370F       0001     000612 370G
     0001     000620 373G       0000     000303 380F       0000     000327 390F       0001     000121 40L        0000     000353 400F
     0000     000455 410F       0000     000466 420F       0001     000741 425G       0000     000502 430F       0000     000517 440F
     0000     000532 450F       0000     000546 460F       0001     000171 50L        0001     001436 572G       0001     001462 576G
     0001     000202 60L        0001     001567 622G       0001     000223 70L        0001     000237 80L        0001     000261 90L
     0000 I   000222 ADD        0000 I   000213 ANC        0000 I   000564 ANCLNG     0000 I   000117 BIT        0003 I   007305 BLK
     0003 I   006155 BLOCK      0003 I   026507 CHNTHS     0003 I   007706 CLD        0003 I   007677 DAS        0003 I   000116 DLMIN
     0000 I   000570 DOI        0000 I   000175 DSL        0000 I   000214 FC         0003 I   000053 FETVEC     0003 I   006011 FLDNAM
     0000 I   000202 FLINE      0003 I   007700 FORMAT     0000 I   000000 FRM        0000 I   000177 FSCAN      0003 I   000000 HEAD
     0000 I   000145 HWRD       0000 I   000163 I          0000 I   000221 IADR       0000 I   000023 IBYTE      0003 I   026510 ICHAIN
     0003 I   026506 ICHN       0000 I   000561 ID         0000 I   000173 IDIM1      0000 I   000174 IDIM2      0000 I   000200 IFLD
     0000 I   000176 IFRST      0000 I   000223 IFT        0000 I   000232 IJ         0000 I   000167 INB        0000     000646 INJP$
     0000     000642 INJP$      0000     000635 INJP$      0000 I   000226 IP         0003 I   007703 IRD        0000 I   000217 IREC
     0000 I   000004 IST        0000 I   000162 ISTAT      0003 I   000111 ISTOP      0000 I   000171 ITEMP      0000 I   000160 IUNIT
     0000 I   000166 IWD        0000 I   000231 IWRDS      0000 I   000224 J          0000 I   000225 JJ         0000 I   000230 JPTS
     0000 I   000061 JREC       0000 I   000216 K          0000 I   000161 KBUF       0000 I   000170 KNB        0003 I   007704 KPTS
     0003 I   000114 KRN        0003 I   026573 KUNIT      0000 I   000215 LC         0000 I   000212 LINC       0000 I   000205 LINEND
     0000 I   000206 LININC     0000 I   000201 LINSTR     0003 I   000112 LNCAT      0000 I   000203 LSKIP      0003 I   007701 MAP
     0003 I   026505 MAXCLS     0000 I   000165 MAXREC     0003 R   000120 MEANS      0000 I   000172 MXSM       0003 I   014612 N
     0000 I   000132 NB         0000 I   000567 NBITS      0003 I   014674 NBLK       0000 I   000565 NC         0000 I   000572 NCAR
     0000 I   000562 NCPR       0000 I   000164 NCS        0000 I   000571 NDSPR      0003 I   000113 NMIN       0003 I   000052 NOFEAT
     0003 I   006010 NOFLD      0003 I   007705 NOPTS      0000 I   000563 NPRC       0000 I   000227 NPTS       0000 I   000561 NRPDS
```

```
0000 I 000566 NS        0000 I 000220 NSAMP     0000 I 000204 NSCAN     0000 I 000233 NWR       0003 I 026504 PUNCH
0000 I 000210 SAMEND    0000 I 000211 SAMINC    0000 I 000207 SAMSTR    0003 I 000117 SEP       0003 I 007702 SBTRIG
0003 I 003054 STDEV     0003 I 000115 STDMAX    0000 I 000573 SVD       0003 I 007615 SYMBLS    0003 I 026572 VARSIZ
```

```
00101       1*      SUBROUTINE TAPERD(IBUF)                                         TAPE0000
00101       2*    C***********************************************************************TAPE0010
00101       3*    C*                                                                TAPE0020
00101       4*    C*    TAPERD READS THE MULTISPECTRAL SCANNER DATA TAPE, UNPACKS THE   TAPE0030
00101       5*    C*    REQUIRED DATA AND STORES IT ON DRUM                         TAPE0040
00101       6*    C*                                                                TAPE0050
00101       7*    C***********************************************************************TAPE0060
00103       8*          INCLUDE COMMON,LIST                                         TAPE0070
00104       8*          PARAMETER MAXPOP=50,MAXFET=30
00105       8*          PARAMETER MAXDIM=25000
00106       8*          PARAMETER MAXFLD=100
00107       8*          COMMON/PASS/HEAD(42),NOFEAT,FETVEC(MAXFET),ISTOP,LNCAT,NMIN,KRN,
00107       8*         * STDMAX,DLMIN,SEP,MEANS(MAXFET,MAXPOP),STDEV(MAXFET,MAXPOP),
00107       8*         * NOFLD,FLDNAM(MAXFLD),BLOCK(MAXFLD,6),BLK(MAXFLD,2),
00107       8*         * SYMBLS(MAXPOP),DAS,FORMAT,MAP,
00107       8*         * SPTRIG,IRD,KPTS,NOPTS,CLD(MAXPOP,MAXPOP)
00107       8*         *, N(MAXPOP),NBLK(MAXFLD,MAXPOP)
00107       8*         * ,PUNCH,MAXCLS,ICHN,CHNTHS,ICHAIN(MAXPOP),VARSIZ
00107       8*         *,KUNIT
00110       8*          INTEGER VARSIZ
00111       8*          INTEGER PUNCH
00112       8*          REAL MEANS
00113       8*          INTEGER SPTRIG,BLOCK,BLK,SYMBLS,HEAD,FETVEC,FLDNAM,DAS
00114       8*          INTEGER FORMAT
00115       8*          END
00116       9*          DIMENSION ID(35),IBUF(1)                                    TAPE0080
00117      10*          IMPLICIT INTEGER(A-Z)                                       TAPE0090
00120      11*          DIMENSION FRM(2,2)                                          TAPE0100
00121      12*          DATA FRM/'UNIVERSAL   ','LARSYS 2'/                         TAPE0110
00123      13*          DIMENSION IST(15),IBYTE(30),JREC(30)                        TAPE0120
00123      14*    C*                                                                TAPE0130
00123      15*    C*    THE ARRAYS BIT,NB,AND HWRD ARE PRECALCULATED WORD AND BIT     TAPE0140
00123      16*    C*    POSITIONS OF INFORMATION IN THE HEADER RECORD OF THE UNIVERSAL  TAPE0150
00123      17*    C*    FORMAT WHICH MUST BE EXTRACTED.                             TAPE0160
00123      18*    C*                                                                TAPE0170
00123      19*    C*      NRPDS - NO. OF RECORDS PER DATA SET                       TAPE0180
00123      20*    C*      NCPR - NO. OF CHANNELS PER RECORD ON RECORDS PAST ANCILLARY RECOTAPE0190
00123      21*    C*      NPRC - NO. OF PHYSICAL RECORDS PER CHANNEL                TAPE0200
00123      22*    C*      ANCLNG - ANCILLARY LENGTH IN BYTES                        TAPE0210
00123      23*    C*      NC - NO. OF CHANNELS                                      TAPE0220
00123      24*    C*      NS - NO. OF SAMPLES PER CHANNEL PER SCAN                  TAPE0230
00123      25*    C*      NBITS - NO. OF BITS PER PIXEL                             TAPE0240
00123      26*    C*      DOI - DATA ORDER INDICATOR                                TAPE0250
00123      27*    C*      NDSPR - NO. OF DATA SETS PER RECORD                       TAPE0260
00123      28*    C*      NCAR - NO. OF CHANNELS OF VIDEO DATA ON SAME RECORD       TAPE0270
00123      29*    C*             WITH ANCILLARY DATA)                               TAPE0280
00123      30*    C*      SVD - START OF VIDEO DATA. (BYTE POSITION WITHIN DATA FOR TAPE0290
00123      31*    C*            A GIVEN CHANNEL)                                    TAPE0300
00123      32*    C*                                                                TAPE0310
00124      33*          DIMENSION BIT(11),NB(11),HWRD(11)                           TAPE0320
00125      34*          DATA HWRD/23,23,23,24,20,397,21,24,395,397,21/              TAPE0330
```

```
00127             DATA BIT/32,16,24,4,28,32,0,20,32,16,8/                        TAPE0340
00131             DATA NB/8,8,8,16,8,16,8,8,8,16,16/                             TAPE0350
00133             EQUIVALENCE (ID(1),NRPDS ),(ID(2),NCPR  ),                     TAPE0360
00133            *            (ID(3),NPRC  ),(ID(4),ANCLNG),                     TAPE0370
00133            *            (ID(5),NC    ),(ID(6),NS    ),                     TAPE0380
00133            *            (ID(7),NBITS ),(ID(8),DOI),                        TAPE0390
00133            *            (ID(9),NDSPR),(ID(10),NCAR ),                      TAPE0400
00133            *            (ID(11),SVD)                                       TAPE0410
00134             DATA IUNIT/3/                                                  TAPE0420
00134       C*                                                                   TAPE0430
00134       C*    READ ID RECORD                                                 TAPE0440
00136             KBUF=680                                                       TAPE0450
00136       C*                                                                   TAPE0460
00137             CALL NTRAN(IUNIT,10)
00140             CALL NTRAN(IUNIT,2,KBUF,IBUF,ISTAT,22)                         TAPE0470
00141             IF (ISTAT.GT.0) GO TO 20                                       TAPE0480
00143          10 WRITE (6,340)                                                  TAPE0490
00145             WRITE (6,400)FRM(1,FORMAT),FRM(2,FORMAT)                       TAPE0500
00151             CALL CMERR                                                     TAPE0510
00152          20 IF (FORMAT.EQ.1) GO TO 40                                      TAPE0520
00154             CALL UNPAK1(IBUF)                                              TAPE0530
00155             DO 30 I=1,6                                                    TAPE0540
00160          30 CALL UNPACK(ID(I),32)                                          TAPE0550
00162             NCS=NC*NS                                                      TAPE0560
00163             MAXREC=(NCS*8 + 32)/36 + 2                                     TAPE0570
00164             NRPDS=1                                                        TAPE0580
00165             NCAR=NC                                                        TAPE0590
00166             ANCLNG=4                                                       TAPE0600
00167             SVD=1                                                          TAPE0610
00170             NBITS=8                                                        TAPE0620
00171             DOI=0                                                          TAPE0630
00172             NCPR=0                                                         TAPE0640
00173             NDSPR=1                                                        TAPE0650
00174             GO TO 100                                                      TAPE0660
00174       C*                                                                   TAPE0670
00174       C*    UNPACK NECESSARY INFORMATION FROM HEADER RECORD-UNIVERSAL FORMAT   TAPE0680
00174       C*                                                                   TAPE0690
00175          40 DO 60 I=1,11                                                   TAPE0700
00200             IWD=HWRD(I)                                                    TAPE0710
00201             IF ((BIT(I)+NB(I)).LE.36) GO TO 50                             TAPE0720
00203             INB=36-BIT(I)                                                  TAPE0730
00204             KNB=NB(I)-INB                                                  TAPE0740
00205             ITEMP=FLD(BIT(I),INB,IBUF(IWD))                                TAPE0750
00206             ID(I)=ITEMP*2**KNB + FLD(0,KNB,IBUF(IWD+1))                    TAPE0760
00207             GO TO 60                                                       TAPE0770
00210          50 CONTINUE                                                       TAPE0780
00211             ID(I)=FLD(BIT(I),NB(I),IBUF(IWD))                              TAPE0790
00212          60 CONTINUE                                                       TAPE0800
00214             MAXREC=680                                                     TAPE0810
00215             IF (NRPDS.LE.15) GO TO 70                                      TAPE0820
00217             WRITE (6,430)NRPDS                                             TAPE0830
00222             CALL CMERR                                                     TAPE0840
00223          70 IF (NPRC.LE.1) GO TO 80                                        TAPE0850
00225             WRITE (6,420)                                                  TAPE0860
00227             CALL CMERR                                                     TAPE0870
00230          80 CONTINUE                                                       TAPE0880
00231             IF(NDSPR.LE.0)NDSPR=1                                          TAPE0890
00233             IF (NBITS.EQ.8) GO TO 90                                       TAPE0900
```

```
00235                 WRITE (6,450)NBITS                                                 TAPE0910
00240                 CALL CHERR                                                         TAPE0920
00241              90 IF (DDI.EQ.0) GO TO 100                                            TAPE0930
00243                 WRITE (6,460)DDI                                                   TAPE0940
00246                 CALL CHERR                                                         TAPE0950
00246           C*                                                                       TAPE0960
00246           C*    DETERMINE MXSM. MAXIMUM NUMBER OF SAMPLES ON ANY PARTICULAR FIELD.TAPE0970
00246           C*    EVERY DATA PT. FROM BEGINNING SAMPLE TO ENDING SAMPLE WILL BE UNPATAPE0980
00246           C*    FOR REQUESTED CHANNELS.                                            TAPE0990
00246           C*                                                                       TAPE1000
00246           C*                                                                       TAPE1010
00247             100 MXSM=0                                                             TAPE1020
00250                 DO 110 I=1,NOFLD                                                   TAPE1030
00253             110 MXSM=MAX(MXSM,(BLOCK(I,5)-BLOCK(I,4)+1))                           TAPE1040
00255                 IDIM1=NRPDS*MAXREC + 1                                             TAPE1050
00256                 IDIM2=IDIM1 + MXSM*NOFEAT +1                                       TAPE1060
00257                 NOPTS=(MAXDIM-IDIM2)/(NOFEAT+1)                                    TAPE1070
00260                 IF (NOPTS.GT.0) GO TO 120                                          TAPE1080
00262                 WRITE (6,390)NC,NS                                                 TAPE1090
00266                 CALL CHERR                                                         TAPE1100
00267             120 CONTINUE                                                           TAPE1110
00270                 KPTS=0                                                             TAPE1120
00271                 IRD=0                                                              TAPE1130
00271           C*                                                                       TAPE1140
00271           C*    DATA SET LENGTH IN BYTES                                           TAPE1150
00272                 DSL=ANCLNG+NS*NC                                                   TAPE1160
00272           C*                                                                       TAPE1170
00272           C*    READ FIRST DATA SET TO DETERMINE FIRST SCAN LINE NUMBER            TAPE1180
00272           C*                                                                       TAPE1190
00273                 CALL BUFILL                                                        TAPE1200
00274                 CALL NTRAN(IUNIT,22)                                               TAPE1210
00275                 IF (IST(1).GT.0) GO TO 130                                         TAPE1220
00277                 WRITE (6,410)IST(1)                                                TAPE1230
00302                 WRITE (6,400)FRM(1,FORMAT),FRM(2,FORMAT)                           TAPE1240
00306                 CALL CHERR                                                         TAPE1250
00307             130 IF(FORMAT.EQ.1)IFRST=FLD(20,16,IBUF(16))                           TAPE1260
00311                 IF(FORMAT.EQ.2)IFRST=FLD(0,16,IBUF(1))                             TAPE1270
00313                 IF (IFRST.GT.0) GO TO 140                                          TAPE1280
00315                 WRITE (6,360)                                                      TAPE1290
00317                 WRITE (6,400)FRM(1,FORMAT),FRM(2,FORMAT)                           TAPE1300
00323                 CALL CHERR                                                         TAPE1310
00324             140 FSCAN=IFRST                                                        TAPE1320
00325                 DO 310 IFLD=1,NOFLD                                                TAPE1330
00330                 LINSTR=BLOCK(IFLD,1)                                               TAPE1340
00331                 IF (LINSTR.GE.IFRST) GO TO 150                                     TAPE1350
00333                 LINSTR=IFRST                                                       TAPE1360
00334                 BLOCK(IFLD,1)=IFRST                                                TAPE1370
00335             150 CONTINUE                                                           TAPE1380
00335           C*    FLINE=FIRST SCAN ON RECORD CONTAINING LINSTR                       TAPE1390
00336                 FLINE=LINSTR-MOD((LINSTR-IFRST),NDSPR)                             TAPE1400
00337                 LSKIP=((FLINE-FSCAN)/NDSPR-1)*NRPDS                                TAPE1410
00337           C*                                                                       TAPE1420
00337           C*    SKIP DOWN THE TAPE TO BEGINNING LINE OF THIS FIELD.                TAPE1430
00337           C*    AND INITIATE READ FOR FIRST DATA SET                               TAPE1440
00337           C*                                                                       TAPE1450
00340                 IF (FSCAN.EQ.FLINE) GO TO 160                                      TAPE1460
00342                 CALL NTRAN(IUNIT,7,LSKIP)                                          TAPE1470
00343                 CALL BUFILL                                                        TAPE1480
```

```
00344            FSCAN=FLINE                                                  TAPE1490
00345        160 CONTINUE                                                     TAPE1500
00346            NSCAN=LINSTR                                                 TAPE1510
00347            IF(BLOCK(IFLD,5).GT.NS)BLOCK(IFLD,5)=NS                      TAPE1520
00351            LINEND=BLOCK(IFLD,2)                                         TAPE1530
00352            LININC=BLOCK(IFLD,3)                                         TAPE1540
00353            SAMSTR=BLOCK(IFLD,4)                                         TAPE1550
00354            SAMEND=BLOCK(IFLD,5)                                         TAPE1560
00355            SAMINC=BLOCK(IFLD,6)                                         TAPE1570
00355      C*    LINC=NO.OF RECORDS TO SKIP AFTER EACH SCCAN LINE             TAPE1580
00356            LINC=(LININC/NDSPR - 1)*NRPDS                                TAPE1590
00357            IF(LINC.LT.0)LINC=0                                          TAPE1600
00357      C*                                                                 TAPE1610
00357      C*    ESTABLISH AREAS ON EACH SCAN LINE TO UNPACK                  TAPE1620
00357      C*                                                                 TAPE1630
00361            ANC=ANCLNG + SAMSTR + SVD - 1                                TAPE1640
00362            IF(FORMAT.EQ.1)ANC=ANC+2                                     TAPE1650
00364            FC=1                                                         TAPE1660
00365            LC=NCAR                                                      TAPE1670
00366            K=1                                                          TAPE1680
00367            DO 210 I=1,NOFEAT                                            TAPE1690
00372            DO 190 IREC=K,NRPDS                                          TAPE1700
00375            IF(IREC.GT.1)ANC=2 + SAMSTR + SVD - 1                        TAPE1710
00377            IF (FETVEC(I).GE.FC.AND.FETVEC(I).LE.LC) GO TO 170           TAPE1720
00401            IF (FETVEC(I).GT.LC.AND.IREC.LT.NRPDS) GO TO 180             TAPE1730
00403            WRITE (6,440)FETVEC(I)                                       TAPE1740
00406            CALL CHERR                                                   TAPE1750
00407        170 IBYTE(I)=(FETVEC(I)-FC)*NS + ANC                             TAPE1760
00410            JREC(I)=IREC                                                 TAPE1770
00411            GO TO 200                                                    TAPE1780
00412        180 FC=LC+1                                                      TAPE1790
00413            LC=LC+NCPR                                                   TAPE1800
00414        190 CONTINUE                                                     TAPE1810
00416        200 K=IREC                                                       TAPE1820
00417        210 CONTINUE                                                     TAPE1830
00417      C*                                                                 TAPE1840
00417      C*    NSAMP - NO. OF SAMPLES TO UNPACK FOR EACH FEATURE IN FETVEC  TAPE1850
00417      C*                                                                 TAPE1860
00421            NSAMP=SAMEND-SAMSTR + 1                                      TAPE1870
00421      C*                                                                 TAPE1880
00421      C*    BEGIN LOOP FOR READING AND UNPACKING DATA IN THIS FIELD      TAPE1890
00421      C*                                                                 TAPE1900
00422        220 IADR=IDIM1                                                   TAPE1910
00423            ADD = (NSCAN-FSCAN)*DSL                                      TAPE1920
00424            DO 250 IFT=1,NOFEAT                                          TAPE1930
00427            J=JREC(IFT)                                                  TAPE1940
00430            JJ=(J-1)*MAXREC + 1                                          TAPE1950
00430      C*                                                                 TAPE1960
00430      C*    CHECK STATUS OF THIS RECORD BEFORE UNPACKING                 TAPE1970
00430      C*                                                                 TAPE1980
00431        230 IF (IST(J).GE.0) GO TO 240                                   TAPE1990
00433            IF (IST(J).EQ.-2) GO TO 280                                  TAPE2000
00435            IF (IST(J).GE.-1) GO TO 230                                  TAPE2010
00437            WRITE (6,350)                                                TAPE2020
00441            WRITE (6,370)NSCAN,IST(J)                                    TAPE2030
00445            WRITE (6,400)FRM(1,FORMAT),FRM(2,FORMAT)                     TAPE2040
00451            GO TO 280                                                    TAPE2050
00451      C*                                                                 TAPE2060
```

```
00451     209*     C*     UNPACK DATA FOR THIS FEATURE                                         TAPE2070
00451     210*     C*                                                                          TAPE2080
00452     211*       240 IP=ADD + IBYTE(IFT)                                                   TAPE2090
00453     212*           CALL UNPCK(IP,IBUF(JJ),NSAMP,IBUF(IADR))                              TAPE2100
00454     213*           IADR=IADR+MXSM                                                        TAPE2110
00455     214*       250 CONTINUE                                                              TAPE2120
00457     215*           IF ((NSCAN+LININC).GT.LINEND) GO TO 290                               TAPE2130
00461     216*           NSCAN=NSCAN+LININC                                                    TAPE2140
00462     217*           IF (NSCAN.LT.(FSCAN+NDSPR)) GO TO 270                                 TAPE2150
00464     218*           FSCAN=FSCAN + NDSPR*(1 + LINC/NRPDS)                                  TAPE2160
00465     219*           CALL NTRAN(IUNIT,7,LINC)                                              TAPE2170
00466     220*           IF (NSCAN.LT.(FSCAN+NDSPR)) GO TO 260                                 TAPE2180
00470     221*           CALL NTRAN(IUNIT,7,1)                                                 TAPE2190
00471     222*           FSCAN=FSCAN+NDSPR                                                     TAPE2200
00472     223*       260 CONTINUE                                                              TAPE2210
00472     224*     C*                                                                          TAPE2220
00472     225*     C*       INITIATE READ FOR NEXT SCAN                                        TAPE2230
00472     226*     C*                                                                          TAPE2240
00473     227*           CALL BUFILL                                                           TAPE2250
00473     228*     C*                                                                          TAPE2260
00473     229*     C*    TRANSFER DATA FOR THIS SCAN LINE                                      TAPE2270
00474     230*       270 CONTINUE                                                              TAPE2280
00475     231*           CALL TRNSFR(IBUF(IDIM1),IBUF(IDIM2),NSAMP,MXSM,NOFEAT)                TAPE2290
00476     232*           GO TO 220                                                             TAPE2300
00476     233*     C*                                                                          TAPE2310
00476     234*     C*    END OF DATA                                                           TAPE2320
00476     235*     C*                                                                          TAPE2330
00477     236*       280 IF(NSCAN.GT.LINSTR)NSCAN=NSCAN-LININC                                 TAPE2340
00501     237*           BLOCK(IFLD,2)=NSCAN                                                   TAPE2350
00502     238*           WRITE (6,380)IFLD,NSCAN                                               TAPE2360
00506     239*           IF (IFLD.EQ.NOFLD) GO TO 300                                          TAPE2370
00506     240*     C*                                                                          TAPE2380
00506     241*     C*    REWIND TAPE AND POSITION AT FIRST SCAN LINE                           TAPE2390
00506     242*     C*                                                                          TAPE2400
00510     243*           CALL NTRAN(IUNIT,22,10,22)                                            TAPE2410
00511     244*           CALL NTRAN(IUNIT,7,1)                                                 TAPE2420
00512     245*           FSCAN=IFRST                                                           TAPE2430
00513     246*           CALL BUFILL                                                           TAPE2440
00514     247*           GO TO 300                                                             TAPE2450
00515     248*       290 CALL TRNSFR(IBUF(IDIM1),IBUF(IDIM2),NSAMP,MXSM,NOFEAT)                TAPE2460
00516     249*       300 BLK(IFLD,1)=(LINEND-LINSTR)/LININC + 1                                TAPE2470
00517     250*           BLK(IFLD,2)=(SAMEND-SAMSTR)/SAMINC + 1                                TAPE2480
00520     251*       310 CONTINUE                                                              TAPE2490
00522     252*            CALL NTRAN(IUNIT,22)                                                 TAPE2500
00522     253*     C*                                                                          TAPE2510
00522     254*     C*    WRITE PARTIALLY FILLED BUFFER ON TO DRUM.  RECALCULATE THE NUMBER TAPE2520
00522     255*     C*    OF POINTS WHICH THE IBUF ( IDAT) ARRAY WILL HOLD (USING THE BUFFERTAPE2530
00522     256*     C*    AFEAS WHICH WERE USED IN TAPE I/O).  IF ALL OF DATA WILL FIT IN COTAPE2540
00522     257*     C*    READ IT BACK INTO IBUF(1) FROM DRUM.  IF NOT, READJUST THE NUMBER TAPE2550
00522     258*     C*     RECORDS AND RECORD LENGTH.  ** WARNING **  THIS CAN BE DONE ONLY TAPE2560
00522     259*     C*    DRUM , BINARY I/O                                                     TAPE2570
00522     260*     C                                                                           TAPE2580
00523     261*           VARSIZ=NOFEAT*(NOFEAT+1)/2                                            TAPE2590
00524     262*           NPTS=(MAXDIM-VARSIZ*MAXCLS)/(NOFEAT+1)                                TAPE2600
00525     263*           JPTS=NOPTS*IRD + KPTS                                                 TAPE2610
00526     264*           IF (KPTS.EQ.0) GO TO 320                                              TAPE2620
00530     265*           IWRDS=NOFEAT*KPTS                                                     TAPE2630
00531     266*           CALL NTRAN(4,1,IWRDS,IBUF(IDIM2),ISTAT,22)                            TAPE2640
```

```
00532          320 IRD=JPTS/NPTS                                                     TAPE2650
00533              KPTS=JPTS-IRD*NPTS                                                TAPE2660
00534              IF(KPTS.GT.0)IRD=IRD+1                                            TAPE2670
00536              NOPTS=NPTS                                                        TAPE2680
00537              IF(IRD.GT.1)RETURN                                                TAPE2690
00541              CALL NTRAN(4,10)                                                  TAPE2700
00542              IWRDS=KPTS*NOFEAT                                                 TAPE2710
00543              CALL NTRAN(4,2,IWRDS,IBUF,ISTAT,22)                               TAPE2720
00544              IRD=0                                                             TAPE2730
00545              RETURN                                                            TAPE2740
00546          330 FORMAT(' END-OF-FILE ENCOUNTERED ON HEADER RECORD')               TAPE2750
00547          340 FORMAT(' UNRECOVERABLE ERROR READING HEADER RECORD')              TAPE2760
00550          350 FORMAT(' ERROR WHILE READING DATA RECORD')                        TAPE2770
00551          360 FORMAT(' A LINE NO. IS LESS THAN OR EQUAL ZERO')                  TAPE2780
00552          370 FORMAT(' LAST SCAN LINE READ',I5,' ISTAT=',I5)                    TAPE2790
00553          380 FORMAT(' FIELD BOUNDARY FOR FIELD',I4,' DEFINED BEYOND SCOPE OF DATAPE2800
00553             *TA'/' THIS FLIGHT LINE CONTAINS',I6,' SCAN LINES')                TAPE2810
00554          390 FORMAT(/' INTERNAL DIMENSIONS TOO SMALL FOR DATA'/' NO. OF CHANNELTAPE2820
00554             *S ON DATA TAPE=',I7,' NO. OF POINTS/CHANNEL=',I7/)                TAPE2830
00555          400 FORMAT(' YOU HAVE INDICATED DATA TAPE IS IN',2A6,' FORMAT'//      TAPE2840
00555             *                                      ' CHECK THE FOLLOWING POSTAPE2850
00555             *SIBLE ERRCRS'/' 1. HAVE YOU INDICATED CORRECT FORMAT '/           TAPE2860
00555             *' 2. IF DATA TAPE IS 9-TRACK, THE -ASG- CARD SHOULD HAVE AN -N- OPTAPE2870
00555             *TION'/4X,'AND A MESSAGE TO OPERATOR SHOULD BE ON 588 FORM'/       TAPE2880
00555             *' 3. IF THE DATA TAPE WAS GENERATED ON A MACHINE OTHER THAN THE 11TAPE2890
00555             *08'/4X,'THE -ASG- CARD SHOULD HAVE AN -A- OPTION')                TAPE2900
00556          410 FORMAT(' ERROR READING FIRST DATA RECORD--ISTAT=',I3)             TAPE2910
00557          420 FORMAT(' ONLY ONE OR LESS RECORDS PER CHANNEL ACCEPTABLE AT THIS TTAPE2920
00557             *IME')                                                             TAPE2930
00560          430 FORMAT(' NO. OF RECORDS PER DATA SET=',I5,' MUST BE LESS THAN OR ETAPE2940
00560             *QUAL 15')                                                         TAPE2950
00561          440 FORMAT(' FEATURE NUMBERS' I5,' AND ABOVE ARE NOT ON DATA TAPE'/   TAPE2960
00561             *)                                                                 TAPE2970
00562          450 FORMAT(' NO. OF BITS/PIXEL=',I5,' ONLY 8 BITS ACCEPTABLE AT THIS TTAPE2980
00562             *IME')                                                             TAPE2990
00563          460 FORMAT(' DATA ORDER INDICATOR=',I5/' DATA MUST BE ORDERED BY PIXELTAPE3000
00563             *')                                                                TAPE3010
00563        C*                                                                      TAPE3020
00563        C*    INTERNAL ROUTINE TO TRANSFER DATA FROM ONE BUFFER AREA TO         TAPE3030
00563        C*    ANOTHER AND DUMP ON TO DRUM WHEN FILLED                           TAPE3040
00563        C*                                                                      TAPE3050
00564              SUBROUTINE TRNSFR(IBUF,CBUF,NSAMP,MXSM,NOFEAT)                    TAPE3060
00567              DIMENSION IBUF(MXSM,NOFEAT),CBUF(NOFEAT,NOPTS)                    TAPE3070
00570              REAL CBUF                                                         TAPE3080
00571              DO 300 IJ=1,NSAMP,SAMINC                                          TAPE3090
00574              KPTS=KPTS+1                                                       TAPE3100
00575              DO 290 J=1,NOFEAT                                                 TAPE3110
00600              CBUF(J,KPTS)=IBUF(IJ,J)                                           TAPE3120
00601          290 CONTINUE                                                          TAPE3130
00603              IF(KPTS.LT.NOPTS)GO TO 300                                        TAPE3140
00605              NWR=KPTS*NOFEAT                                                   TAPE3150
00606              CALL NTRAN(4,1,NWR,CBUF,ISTAT,22)                                 TAPE3160
00607              KPTS=0                                                            TAPE3170
00610              IRD=IRD+1                                                         TAPE3180
00611          300 CONTINUE                                                          TAPE3190
00613              RETURN                                                            TAPE3200
00613        C*                                                                      TAPE3210
00613        C*    INTERNAL ROUTINE TO INITIATE READS FOR ONE SCAN LINE)             TAPE3220
```

```
00614     325*          SUBROUTINE BUFILL                                                  TAPE3230
00617     326*          CALL NTRAN(IUNIT,22)                                               TAPE3240
00620     327*          K=1                                                                TAPE3250
00621     328*          DO 310 I=1,NRECS                                                   TAPE3260
00624     329*          CALL NTRAN(IUNIT,2,MAXREC,IBUF(K),IST(I))                          TAPE3270
00625     330*          K=K+MAXREC                                                         TAPE3280
00626     331*      310 CONTINUE                                                           TAPE3290
00630     332*          RETURN                                                             TAPE3300
00631     333*          END                                                                TAPE3310

     END OF COMPILATION:        NO  DIAGNOSTICS.
   TAPERD          SYMBOLIC                            29 NOV 73  13:41:14   0  01622072    14   333  (DELETED)
   TAPERD  CODE    RELOCATABLE                         29 NOV 73  13:41:14   1  01633160    36     1  (DELETED)
                                                                             0  01633224    14   121
```

```
@ ASM,* TDATE,TDATE                                                                          03 DEC 73          12:23:16.850
ASSEMBLED BY UNIVAC 1108 EXEC II ASSEMBLER   2404 0008A
THIS ASSEMBLY WAS DONE ON 03 DEC 73 AT 12:23:15
000001                                                 .  IDENTIFICATION
000002                                                 .    TITLE          TODAYS  DATE
000003                                                 .    PROGRAMMER     LOCKHEED ELECTRONICS  TEST DATA REDUCTION
000004                                                 .    DATE           OCT 1967
000005                                                 .    INSTALLATION   MANNED SPACECRAFT CENTER  -  HOUSTON
000006                                                 .  DESCRIPTION
000007                                                 .    THIS ROUTINE IS DESIGNED TO BE CALLED BY FORTRAN V
000008                                                 .    AND TO RETURN 8 CHARACTER S CONTAINING   DAY MONTH YEAR   XX/XX/XX
000009                                                 .
000010                                                 .
000011                                                                     NBPR
000012                                                 TDATE*   .
000013                                                 ITDATE*  .
000014      00    000000  06 01 13 00 0 000006                  S,1       B11,B11H
000015            000001  27 00 14 13 0 000000                  L         A0,*0,B11
000016            000002  24 16 14 00 0 000001                  A,14      A0,+1
000017            000003  06 01 14 00 0 000005                  S,1       A0,S+2
000018                                                 F        FORM      6,12,18
000019 W          000004  74 13 13 00 0 000000                  LMJ       11,EDATES
000020            000005  00 0002 000000                        F         0,2,S-S
000021            000006  27 16 13 00 0 000000         B11H     L,14      B11,S-S
000022            000007  74 04 00 13 0 000002                  J         2,11
000023                                                          END
                                                       EDATES
   TDATE             SYMBOLIC                                                 29 NOV 73  13:41:16   0  01636462    14    23  (DELETED)
   TDATE    CODE     RELOCATABLE                                              29 NOV 73  13:41:16   1  01637164    12     1  (DELETED)
                                                                                                    0  01637200    14     1
```

ORIGINAL PAGE IS
OF POOR QUALITY

```
@ FOR,* TWRITE,TWRITE                                                                    03 DEC 73          12:23:17.574
UNIVAC 1108 FORTRAN V EXEC II LEVEL 25A  -(EXEC8 LEVEL E1201001OA)
THIS COMPILATION WAS DONE ON 03 DEC 73 AT 12:23:17

   SUBROUTINE TWRITE    ENTRY POINT 000474

   STORAGE USED: CODE(1) 000510; DATA(0) 000226; BLANK COMMON(2) 000000

    COMMON BLOCKS:

    0003    PASS    026574

   EXTERNAL REFERENCES (BLOCK, NAME)

    0004    NNCOD$
    0005    NTRAN
    0006    NI01$
    0007    NI02$
    0010    NERR3$

   STORAGE ASSIGNMENT  (BLOCK, TYPE, RELATIVE LOCATION, NAME)

    0001    000301 115L      0001    000010 123G      0001    000415 125L      0001    000014 130G      0001    000032 141G
    0001    000056 151G      0001    000063 156G      0001    000117 167G      0001    000131 175G      0001    000143 205G
    0001    000156 211G      0001    000220 227G      0001    000227 234G      0001    000241 242G      0001    000045 25L
    0001    000253 252G      0001    000266 256G      0001    000325 274G      0001    000360 307G      0001    000402 313G
    0001    000101 35L       0000    000161 500F      0000    000163 510F      0000    000164 520F      0001    000171 65L
    0000  I 000001 BCDRAY    0003  I 007305 BLK       0003  I 006155 BLOCK     0003    026507 CHNTHS    0000  R 000040 CHNVEC
    0003    007706 CLD       0003  I 007677 DAS       0003    000116 DLMIN     0003  I 000053 FETVEC    0003  I 006011 FLDNAM
    0003  I 007700 FORMAT    0003  I 000000 HEAD      0000  I 000134 I         0000  I 000000 IBLANK    0003    026510 ICHAIN
    0003    026506 ICHN      0000  I 000076 ICHVEC    0000    000174 INJPS     0003    007703 IRD       0000  I 000137 ISTAT
    0003    000111 ISTOP     0000  I 000141 I1        0000  I 000143 I2        0000  I 000146 I3        0000  I 000147 I4
    0000  I 000135 J         0000  I 000155 J1        0000  I 000157 J2        0000  I 000160 J3        0000  I 000140 K
    0000  I 000136 KA        0003    007704 KPTS      0003    000114 KRN       0003  I 026573 KUNIT     0000  I 000150 K1
    0000  I 000151 K2        0000  I 000152 K3        0000  I 000153 K4        0000  I 000154 K5        0000  I 000144 L
    0003  I 000112 LNCAT     0000  I 000145 L1        0003    007701 MAP       0003    026505 MAXCLS    0003  R 000120 MEANS
    0003  I 014612 N         0003    014674 NBLK      0003    000113 NMIN      0003  I 000052 NOFEAT    0000  I 000142 NOFEA1
    0003    006010 NOFLD     0003    007705 NOPTS     0000  I 000156 NOREC     0003  I 026504 PUNCH     0003    000117 SEP
    0003  I 007702 SPTRIG    0003  R 003054 STDEV     0003    000115 STOMAX    0003  I 007615 SYMBLS    0003  I 026572 VARSIZ
    0000  I 000037 WORDLN


00100        1*      C****                                                                 ****
00100        2*      C*       WRITES ON TAPE IN BCD FORM STATISTICS FROM ISOSCLS TO BE READ       *
00100        3*      C*                              BY ERIPS                                       *
00100        4*      C****                                                                 ****
00101        5*               SUBROUTINE TWRITE(COVAR)
00103        6*               INCLUDE COMMON,LIST
00104        6*               PARAMETER MAXPOP=50,MAXFET=30
00105        6*               PARAMETER MAXDIM=25000
00106        6*               PARAMETER MAXFLD=100
00107        6*               COMMON/PASS/HEAD(42),NOFEAT,FETVEC(MAXFET),ISTOP,LNCAT,NMIN,KRN,
00107       6*          * STDMAX,DLMIN,SEP,MEANS(MAXFET,MAXPOP),STDEV(MAXFET,MAXPOP),
00107       6*          * NOFLD,FLDNAM(MAXFLD),BLOCK(MAXFLD,6),BLK(MAXFLD,2),
00107       6*          * SYMBLS(MAXPOP),DAS,FORMAT,MAP,
00107       6*          * SPTRIG,IFD,KPTS,NOPTS,CLD(MAXPOP,MAXPOP)
00107       6*          *, N(MAXPOP),NBLK(MAXFLD,MAXPOP)
00107       6*          * ,PUNCH,MAXCLS,ICHN,CHNTHS,ICHAIN(MAXPOP),VARSIZ
00107       6*          *,KUNIT
00110       6*            INTEGER VARSIZ
00111       6*            INTEGER PUNCH
00112       6*            REAL MEANS
00113       6*            INTEGER SPTRIG,BLOCK,BLK,SYMBLS,HEAD,FETVEC,FLDNAM,DAS
00114       6*            INTEGER FORMAT
00115       6*            END
00116       7*            DATA IBLANK/'      '/
00120       8*            INTEGER          BCDRAY, WORDLN
00121       9*            DIMENSION BCDRAY(30), CHNVEC(30),COVAR(LNCAT,VARSIZ),ICHVEC(30)
00121      10*      C
00121      11*      C     CHNVEC CONTAINS REREQUESTED FEATURES: 0 -- CHANNEL WAS NOT REQUESTED
00121      12*      C                                           1 -- CHANNEL WAS REQUESTED
00121      13*      C
00122      14*            DO 10 I = 1, 30
00125      15*         10 ICHVEC(I) = 0
00127      16*            DO 20 J = 1, NOFEAT
00132      17*            KA = FETVEC(J)
00133      18*         20 ICHVEC(KA) = 1
00133      19*      C
00133      20*      C     RECORD 1 -- NO. OF CLUSTERS, NO. OF FEATURES, AND REQUESTED CHANNELS
00133      21*      C
00135      22*            ENCODE(500, BCDRAY) LNCAT,NOFEAT,ICHVEC
00145      23*            CALL NTRAN(KUNIT,1, 6,BCDRAY,ISTAT)
00146      24*         25 IF(ISTAT .EQ. 1)  GO TO 25
00150      25*            DO 30   K=1,30
00153      26*         30 BCDRAY(K) = IBLANK
00155      27*            DO 40   I = 1,LNCAT
00155      28*      C
00155      29*      C     RECORD 2 -- NO. OF DATA PTS. IN CLUSTER(I)
00155      30*      C
00160      31*             ENCODE(510,BCDRAY)N(I)
00163      32*            CALL NTRAN(KUNIT,1,3,BCDRAY,ISTAT)
00164      33*         35 IF (ISTAT .EQ. 1)  GO TO 35
00164      34*      C
00164      35*      C     RECORD 3 THRU 7 -- MEANS
00164      36*      C
00166      37*            DO 50   I1 = 1,NOFEAT
00171      38*         50 CHNVEC(I1) =MEANS(I1,I)
00173      39*            NOFEA1  = NOFEAT + 1
00174      40*            DO 60  I2 = NOFEA1 ,30
00177      41*         60 CHNVEC(I2) = 0
00201      42*            L = 1
00202      43*            L1 = 8
00203      44*            WORDLN = 20
00204      45*            DO 70   I3 = 1,4
00207      46*            ENCODE(520,BCDRAY)(CHNVEC(I4), I4=L,L1)
00215      47*            CALL NTRAN(KUNIT,1,WORDLN,BCDRAY,ISTAT)
00216      48*         65 IF(ISTAT .EQ. 1)   GO TO 65
00220      49*            L = L1 + 1
00221      50*            L1 = L1 + 8
00222      51*            IF (I3 .EQ. 3)   WOPDLN = 15
```

ORIGINAL PAGE IS
OF POOR QUALITY

```
00224      52*        70 CONTINUE
00224      53*     C
00224      54*     C       RECORD 8 THRU 11 -- STD. DEV.
00224      55*     C
00226      56*           DO 80    K1 = 1,30
00231      57*        80 BCDRAY(K1) = IBLANK
00233      58*           DO 90    K2 = 1,NOFEAT
00236      59*        90 CHNVEC(K2) = STDEV(K2,I)
00240      60*           NOFEA1  = NOFEAT + 1
00241      61*           DO 100    K3 = NOFEA1 ,30
00244      62*       100 CHNVEC(K3) = 0
00246      63*           WORDLN = 20
00247      64*           L = 1
00250      65*           L1 = 8
00251      66*           DO 110    K4 = 1,4
00254      67*           ENCODE(520,BCDRAY) (CHNVEC(K5), K5=L,L1)
00262      68*           CALL NTRAN(KUNIT,1,WORDLN,BCDRAY,ISTAT)
00263      69*       115 IF(ISTAT .EQ. 1)  GO TO 115
00265      70*           L = L1 + 1
00266      71*           L1 = L1 + 8
00267      72*           IF(K4 .EQ. 3)  WORDLN = 15
00271      73*       110 CONTINUE
00271      74*     C
00271      75*     C       RECORD 13 TRU -- COVARIANCE
00271      76*     C
00273      77*           DO 120    J1 = 1,30
00276      78*       120 BCDRAY(J1) = IBLANK
00300      79*           WORDLN = 20
00301      80*           L = 1
00302      81*           L1 = 8
00303      82*           NOREC = VARSIZ / 8
00304      83*           IF(MOD(VARSIZ,8) .NE. 0)  NOREC = NOREC + 1
00306      84*           DO 130    J2 = 1,NOREC
00311      85*           ENCODE(520,BCDRAY) (COVAR(I,J3),J3=L,L1)
00317      86*           CALL NTRAN(KUNIT,1,WORDLN,BCDRAY,ISTAT)
00320      87*       125 IF(ISTAT .EQ. 1)  GO TO 125
00322      88*           L = L1 + 1
00323      89*           L1 = L1 + 8
00324      90*           IF( J2 .EQ. (NOREC - 1) )  WORDLN = (MOD(VARSIZ,8)*15)/6
00326      91*       130 CONTINUE
00330      92*        40 CONTINUE
00332      93*           CALL NTRAN(KUNIT,22)
00333      94*           RETURN
00334      95*       500 FORMAT(I2,I2,30I1)
00335      96*       510 FORMAT(I6)
00336      97*       520 FORMAT(8E15.7)
00337      98*           END

      END OF COMPILATION:          NO  DIAGNOSTICS.
   TWRITE           SYMBOLIC                                29 NOV 73   13:41:18    0  01637216     14     98  (DELETED)
   TWRITE   CODE    RELOCATABLE                             29 NOV 73   13:41:18    1  01641752     24      1  (DELETED)
                                                                                    0  01642002     14     36
```

```
@ FOR,* UNPACK,UNPACK                                                                                     03 DEC 73              12:23:19.
UNIVAC 1108 FORTRAN V EXEC II LEVEL 25A  -(EXEC8 LEVEL E12010010A)
THIS COMPILATION WAS DONE ON 03 DEC 73 AT 12:23:19

    SUBROUTINE UNPAK1     ENTRY POINT 000202
               UNPACK     ENTRY POINT 000207

    STORAGE USED: CODE(1) 000232; DATA(0) 000036; BLANK COMMON(2) 000000

    EXTERNAL REFERENCES (BLOCK, NAME)

     0003    N4DU$
     0004    NI02$
     0005    NERR3$

    STORAGE ASSIGNMENT  (BLOCK, TYPE, RELATIVE LOCATION, NAME)

     0001    000061 10L        0001    000124 20L        0001    000152 30L        0001    000155 40L        0000    000011 50F
     0000 I  000010 I          0000    000024 INJP$      0000 I  000000 JBIT       0000 I  000005 JMAX       0000 I  000001 KBIT
     0000 I  000002 KWRD       0000 I  000003 KWRD1      0000 I  000007 L          0000 I  000004 NAV        0000 I  000006 NOV


00101        1*            SUBROUTINE UNPAK1 (IDAT)                                                    UNPA0000
00101        2*      C                                                                                 UNPA0010
00101        3*      C     THIS ROUTINE UNPACKS UP TO 32-BIT IBM 360 WORDS INTO                        UNPA0020
00101        4*      C     36-BIT WORD IMAGES WHICH CAN BE READ BY THE UNIVAC 1108.                    UNPA0030
00101        5*      C                                                                                 UNPA0040
00103        6*            DIMENSION IDAT(1)                                                           UNPA0050
00103        7*      C                                                                                 UNPA0060
00103        8*      C     INITIALIZE IDAT ARRAY                                                       UNPA0070
00103        9*      C                                                                                 UNPA0080
00104       10*            JBIT = 0                                                                    UNPA0090
00105       11*            KBIT = -1                                                                   UNPA0100
00106       12*            KWRD = 0                                                                    UNPA0110
00107       13*            KWRD1 = 0                                                                   UNPA0120
00110       14*            NAV = 36                                                                    UNPA0130
00111       15*            RETURN                                                                      UNPA0140
00111       16*      C                                                                                 UNPA0150
00112       17*            ENTRY UNPACK (NTRE, NBITS)                                                  UNPA0160
00112       18*      C                                                                                 UNPA0170
00112       19*      C     UNPACK NEXT NBITS FROM IREC ARRAY INTO NTRE                                 UNPA0180
00112       20*      C                                                                                 UNPA0190
00114       21*            IF (NBITS.GT.32) GO TO 40                                                   UNPA0200
00116       22*            IF (NBITS.LE.0) GO TO 30                                                    UNPA0210
00116       23*      C                                                                                 UNPA0220
00120       24*            KBIT = KBIT + NBITS                                                         UNPA0230
00121       25*            KWRD = KBIT / 36                                                            UNPA0240
00122       26*            IF (NAV.LT.NBITS) GO TO 10                                                  UNPA0250
00124       27*            NTRE = FLD(JBIT,NBITS,IDAT(KWRD+1))                                         UNPA0260
00125       28*            JBIT = JBIT + NBITS                                                         UNPA0270
00126       29*            NAV = NAV - NBITS                                                           UNPA0280
00127       30*            KWRD1 = KWRD                                                                UNPA0290
```

```
00130      31*           RETURN                                                          UNPA0300
00130      32*      C                                                                    UNPA0310
00131      33*        10 JMAX = (KWRD1 + 1) * 36 - 1                                     UNPA0320
00132      34*           NOV = KBIT - JMAX                                               UNPA0330
00133      35*           L = 36 - NBITS                                                  UNPA0340
00134      36*           IF (NAV.LE.0) GO TO 20                                          UNPA0350
00136      37*           I = 36 - NAV                                                    UNPA0360
00137      38*           FLD(L,NAV,NTRE) = FLD(I,NAV,IDAT(KWRD))                         UNPA0370
00140      39*           L = L + NAV                                                     UNPA0380
00140      40*      C                                                                    UNPA0390
00141      41*        20 FLD(L,NOV,NTRE) = FLD(ABS(0),NOV,IDAT(KWRD+1))                  UNPA0400
00142      42*           JBIT = NOV                                                      UNPA0410
00143      43*           KWRD1 = KWRD                                                    UNPA0420
00144      44*           NAV = 36 - NOV                                                  UNPA0430
00144      45*      C                                                                    UNPA0440
00145      46*        30 RETURN                                                          UNPA0450
00145      47*      C                                                                    UNPA0460
00146      48*        40 WRITE (6,50)NBITS                                               UNPA0470
00151      49*        50 FORMAT (1H ,'ERROR.',I4,' BITS EXCEEDS 32-BIT WORD LENGTH.')    UNPA0480
00152      50*           GO TO 30                                                        UNPA0490
00153      51*           END                                                             UNPA0500


     END OF COMPILATION:          NO  DIAGNOSTICS.
   UNPACK          SYMBOLIC                                        29 NOV 73  13:41:19   0  01642772    14    51  (DELETED)
   UNPACK  CODE    RELOCATABLE                                     29 NOV 73  13:41:19   1  01644304    24     1  (DELETED)
                                                                                         0  01644334    14    18
```

```
@ ASM,• UNPCK,UNPCK                                                                                             03 DEC 73            12:23:21•
ASSEMBLED BY UNIVAC 1108 EXEC II ASSEMBLER    2404 0008A
THIS ASSEMBLY WAS DONE ON 03 DEC 73 AT 12:23:21
000001                                                       •                                                                  UNPK0010
000002                                                       • CALL UNPCK (ISTART,STRINGI,LENGTH,STRINGO)                       UNPK0020
000003                                                       • ISTART  - LOCATION OF FIRST DATA BYTE IN THE ARRAY STRINGI       UNPK0030
000004                                                       • STRINGI - INPUT ARRAY CONTAINING 8 BIT/BYTE DATA IN PACKED FORMAT UNPK0040
000005                                                       • LENGTH - NUMBER OF BYTES TO BE UNPACKED                          UNPK0050
000006                                                       • STRINGO - OUTPUT ARRAY CONTAINING THE UNPACKED DATA              UNPK0060
000007                                                       •                                                                  UNPK0070
000008                                                       • UNPACKING PERFORMAED IN 2 WORD (9BYTE) GROUPINGS OR PACKETS      UNPK0080
000009                                                       •                                                                  UNPK0090
000010                                                      $(1)                        •                                       UNPK0140
000011                                                              NBPR                •                                       UNPK0150
000012     .01   000000  74 06 00 00 0 000002                       NOP      0:NAME     •                                       UNPK0160
000013           000001  06 01 13 00 0 000003               UNPCK•  SX,H2    X11,NAME+1 •                                       UNPK0170
000014           000002  71 12 04 00 0 000010                       DS       A4,SAVEA   •                                       UNPK0180
000015           000003  01 00 06 00 0 000012                       S        A6,SAVEA+2 •                                       UNPK0190
000016           000004  06 00 01 00 0 000006                       SX       X1,SAVEX   •                                       UNPK0200
000017           000005  06 00 02 00 0 000007                       SX       X2,SAVEX+1 •                                       UNPK0210
000018           000006  04 00 01 00 0 000013                       SR       R1,SAVER   •                                       UNPK0220
000019           000007  04 00 02 00 0 000014                       SR       R2,SAVER+1 •                                       UNPK0230
000020           000010  10 00 04 13 1 000000                       L        A4,*0,X11  •                                       UNPK0240
000021           000011  05 00 00 00 0 000017                       SZ       A3         •                                       UNPK0250
000022           000012  34 16 03 00 0 000011                       DI,U     A3,9       •                                       UNPK0260
000023           000013  74 01 04 00 0 000016                      JNZ      A4,$+3
000024           000014  25 16 17 00 0 000001                      AN,U     A3,1
000025           000015  14 16 04 00 0 000011                      A,U      A4,9
000026           000016  24 00 17 00 0 000017                       A        A3,A3      •                                       UNPK0270
000027           000017  24 16 17 13 1 000001                       A,U      A3,*1,X11  • A3 CONTAINS ABS. LOC. OF FIRST PACKET     UNPK0280
000028           000020  24 00 17 00 0 000015                       A        A3,(2,0)   • MODIFIED FOR INDEX INCREMENTATION         UNPK0290
000029           000021  27 00 01 00 0 000017                       LX       X1,A3      • X1 CONTAINS THE ABS. LOC. INPUT PACKET    UNPK0300
000030           000022  31 16 04 00 0 000010                       MSI,U    A4,8       • A4 CONTAINS BIT SHIFT FOR BYTE IN PACKE   UNPK0310
000031           000023  71 13 00 01 2 000000                       DL       A0,0,*X1   • A0,A1 CONTAINS THE 2 WORD PACKET          UNPK0320
000032           000024  27 00 02 00 0 000020                       LX       X2,A4      •                                       UNPK0330
000033           000025  73 11 00 02 0 000000                       LDSC     A0,0,X2    •                                       UNPK0340
000034           000026  23 00 02 00 0 000004                       LR       R2,MASK    •                                       UNPK0350
000035           000027  27 00 16 13 1 000002                       L        A2,*2,X11  •                                       UNPK0370
000036           000030  25 16 16 00 0 000001                       AN,U     A2,1       •                                       UNPK0380
000037           000031  23 00 01 00 0 000016                       LR       R1,A2      • R1 CONTAINS LENGTH -1                     UNPK0390
000038           000032  27 16 02 13 1 000003                       LX,U     X2,*3,X11  • X2 LOCATION OF OUTPUT WORD MODIFIED FOR   UNPK0400
000039           000033  24 00 02 00 0 000016                       AX       X2,(1,0)   •     INDEX INCREMENTATION                  UNPK0410
000040           000034  43 00 01 00 0 000017               LOOP    MLU      A1,(0)     • A2 CONTAINS ANSWER                        UNPK0420
000041           000035  06 00 16 02 2 000000                       S        A2,0,*X2   • STORE IN OUTPUT ARRAY                     UNPK0460
000042           000036  14 16 04 00 0 000010                       A,U      A4,8       •                                       UNPK0500
000043           000037  53 00 04 00 0 000005                       TNE      A4,NW      • IF A4 LESS THAN 80 CONTINUE               UNPK0510
000044           000040  74 04 00 00 0 000044                       J        LOADP      • IF A4 EQUAL TO 80 LOAD NEXT PACKET        UNPK0520
000045           000041  73 11 00 00 0 000010               CONT    LDSC     A0,8       •                                       UNPK0530
000046           000042  70 04 01 00 0 000034                       JGD      R1,LOOP    •                                       UNPK0540
000047           000043  74 04 00 00 0 000047                       J        RETRN      •                                       UNPK0550
000048           000044  71 13 00 01 2 000000               LOADP   DL       A0,0,*X1   • LOAD NEXT WORD PACKET                     UNPK0560
000049           000045  10 16 04 00 0 000010                       L,U      A4,8       •                                       UNPK0570
000050           000046  74 04 00 00 0 000041                       J        CONT       •                                       UNPK0580
000051           000047  23 00 03 00 0 000051               RETRN   LR       R3,$+2     •                                       UNPK0590
000052           000050  74 04 00 00 0 000052                       J        $+2        •                                       UNPK0600
000053           000051  000050 000032                               +       $-UNPCK,NAME
000054           000052  27 00 13 00 0 000003                       LX       X11,NAME+1 •                                       UNPK0620
000055           000053  71 13 04 00 0 000010                       DL       A4,SAVEA   •                                       UNPK0630
```

```
000056         000054  10 00 06 00 0 000012            L        A6,SAVEA+2      .              UNPK0640
000057         000055  27 00 01 00 0 000006            LX       X1,SAVEX        .              UNPK0650
000058         000056  27 00 02 00 0 000007            LX       X2,SAVEX+1      .              UNPK0660
000059         000057  23 00 01 00 0 000013            LR       R1,SAVER        .              UNPK0670
000060         000060  23 00 02 00 0 000014            LR       R2,SAVER+1      .              UNPK0680
000061         000061  50 15 00 00 1 000000            TZ,S1    *NAME-2         .              UNPK0690
000062         000062  74 04 00 00 1 000030            J        *NAME-2         .              UNPK0700
000063 U       000063  74 04 00 00 0 000000   ENDR     J        NERRS           .              UNPK0710
000064                                        $(0)'                             .              UNPK0720
000065     00  000000  000013 000005                   +        11,5            .              UNPK0730
000066         000001  000000000000                    +        0               .              UNPK0740
000067         000002  003223251023           NAME     +        'UNPCK'
000068         000003  000000000000                    +        S-S             .              UNPK0760
000069         000004  777777777400           MASK     +        0777777777400   .              UNPK0770
000070         000005  000000000120           N8       +        80              .              UNPK0780
000071         000006                         SAVEX    RES      2               .              UNPK0790
000072         000010                         SAVEA    RES      3               .              UNPK0800
000073         000013                         SAVER    RES      2               .              UNPK0810
000074                                                 END                      .              UNPK0820
               000015  000002 000000
               000016  000001 000000
               000017  000000000000
                                              NERRS
    UNPCK           SYMBOLIC                                              29 NOV 73  13:41:21  0  01644730   14    74  (DELETED)
    UNPCK    CODE   RELOCATABLE                                           29 NOV 73  13:41:21  1  01646744   12     1  (DELETED)
                                                                                               0  01646760   14     7
```

5.5 Sample Input

```
@Z RUN L73179,TF4,H3,1659J,Q619,C,5,5                      MINTER
@N MSG      FILE REQ.  TAPE 1  FH432      FSTRN
@ TCP
@  ASG A=A09654
@ ASG C=A03030
@ XQT CUR
  IN A
@ XQT ISOCLS
FEATURES  1,4,7,9
PUNCH
SYMBOLS   /,*,.,+,&,M,-
KRN       10
NMIN       15
MAXCLS    10
*END*
FLD 1                340  400    2    1  220    2
FLD 2                644  694    2    1  220    2
$END*
@E PMD
```

5.6 Sample Output

```
INPUT SUMMARY

     FEATUR    1,4,7,9
     PUNCH
     SYMBOL    /,*,.,:,+,$,#,-
     KMN       10
     NMIN       15
     MAXCLS    10
     *END*


                        FIELDS TO BE CLUSTERED

         FIELD      FIRST     LAST      LINE      FIRST     LAST      SAMPLE
         NAME       LINE      LINE      INC.      SAMPLE    SAMPLE    INC.

     1.  FLD 1       340       400         2         1       220         2
     2.  FLD 2       644       694         2         1       220         2
```

```
COVARIANCE MATRIX FOR CLUSTER    1

          5.31
          2.19      2.87
          1.52      2.26      5.92
          2.15      3.17      3.75      7.60


COVARIANCE MATRIX FOR CLUSTER    2

         10.47
          2.10      2.03
         -1.29      1.49      5.05
          1.16      2.35      2.17      8.22


COVARIANCE MATRIX FOR CLUSTER    3

         19.84
          5.20      3.59
           .22      1.74      6.97
          -.61      1.46      2.85      8.06


COVARIANCE MATRIX FOR CLUSTER    4
```

```
               4.33
               1.48     3.39
               1.65     3.79     9.52
               1.10     5.51    10.13    26.59


      COVARIANCE MATRIX FOR CLUSTER    5

               5.69
               1.17     2.06
               1.27     1.93     9.97
              -1.40     1.30     2.62    14.35


      COVARIANCE MATRIX FOR CLUSTER    6

              98.89
              10.76    22.92
               1.56    18.82    26.00
               9.60    28.66    35.51    54.27


      COVARIANCE MATRIX FOR CLUSTER    7

              14.97
               3.32     3.25
               1.13     2.02     3.42
               2.36     3.36     3.66    10.27
```

```
COVARIANCE MATRIX FOR CLUSTER   8

      11.54

       5.05    16.92

       7.82    20.33    34.24

       7.17    20.60    34.82    60.71



COVARIANCE MATRIX FOR CLUSTER   9

     217.20

      96.30    65.01

      59.12    45.19    47.78

      46.86    45.22    56.49    63.98



COVARIANCE MATRIX FOR CLUSTER  10

       5.02

       1.11     2.03

       1.64     1.19     2.69

      -1.03     1.72     1.41     8.53
```

```
                                        MANNED SPACECRAFT CENTER                        07 NOV 72
                                            HOUSTON, TEXAS



        TOTAL NUMBER OF CLUSTERS = 10

        TOTAL NUMBER OF POINTS =    6270

        CLUSTER      SYMBOL      POINTS IN CLUSTER
           1            /               1171
           2            +               1290
           3            .                490
           4            *                382
           5            A                430
           6            #                 44
           7            -                996
           8            8                194
           9            9                 50
          10            A               1223


                     MEANS

        CLUSTER          CH( 1)    CH( 4)    CH( 7)    CH( 9)
           1             181.79    200.88    197.17    200.35
           2             167.58    191.35    187.96    169.74
           3             175.54    196.72    193.23    192.92
           4             175.30    189.58    180.70    160.68
           5             179.93    194.31    189.46    172.02
           6             149.18    175.91    174.50    160.02
           7             163.42    188.13    184.99    177.82
           8             171.97    183.29    170.23    145.23
           9             120.36    151.92    146.86    118.30
          10             182.60    197.04    193.13    179.48


                STANDARD DEVIATIONS

        CLUSTER          CH( 1)    CH( 4)    CH( 7)    CH( 9)
           1               2.23      1.58      2.36      2.69
           2               3.20      1.56      2.16      2.80
           3               4.45      1.65      2.61      2.81
           4               2.06      1.80      3.07      5.15
           5               2.37      1.39      2.20      3.78
           6               9.94      4.79      5.10      7.37
           7               3.85      1.72      1.77      3.18
           8               3.39      4.11      5.85      7.79
           9              14.74      8.06      6.91      9.16
```

```
      10            2.16    1.20     1.52     2.88


                   DISTANCES BETWEEN CLUSTERS

      CLUSTER      1      2      3      4      5      6      7      8      9     10
         1        .00  10.69   4.44  14.33  10.53  15.98  13.92  16.07  27.05   8.25
         2      10.69    .00   5.28   7.64   6.32   9.42   3.60  10.42  21.37   7.78
         3       4.44   5.28    .00  10.33   6.91  11.96   8.48  12.77  22.72   5.25
         4      14.33   7.64  10.33    .00   5.59   7.58   6.26   4.36  17.01   9.65
         5      10.53   6.32   6.91   5.59    .00  10.78   7.30   9.07  21.56   3.84
         6      15.98   9.42  11.96   7.58  10.78    .00   7.02   4.75   8.25  13.69
         7      13.92   3.60   8.48   6.26   7.30   7.02    .00   8.52  19.19  10.24
         8      16.07  10.42  12.77   4.36   9.07   4.75   8.52    .00  10.33  12.75
         9      27.05  21.37  22.72  17.01  21.56   8.25  19.19  10.33    .00  25.78
        10       8.25   7.78   5.25   9.65   3.84  13.69  10.24  12.75  25.78    .00
```

```
                                   MANNED SPACECRAFT CENTER                                07 NOV 72
                                       HOUSTON, TEXAS



       FLD 1

       TOTAL NUMBER OF POINTS IN THIS FIELD    3410
       LINE  340 THROUGH  400 EVERY  2 LINE(S)    SAMPLE    1 THROUGH 220 EVERY  2 SAMPLE(S)


             00000000000000000000000000000000000000000000000000111111111111111111111111111111111111111111111111112222222222
             00000111112222233333444445555566666777778888899999000001111122222333334444455555666667777788888999990000011111
             13579135791357913579135791357913579135791357913579135791357913579135791357913579135791357913579135791357913579

       340   ......-.....,.......*****************************......8-AAAAAA..//////.A.AAAA..A..,.**-8999999#*A,AAAAAAAAAAA../*/
       342   ....**********.,.....**************--*******-****,.***-*#*A.......A,..A-*8999999#*AAAAAAAAAAAAAAAAAAAAAAAAAAA.....
       344   ,.......*...*.......**-*************---*****-*****,*****-89999999#*.A.AAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAA...../
       346   /.......*...*.....*.***-*************-*******-#999999#-.-8#.AAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAA.....
       348   /.......***.........*------*----#999999-***.****,*****.#.AAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAA.....
       350   /.....**********-/*-#99999#*************************-AAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAA.....
       352   /....*****------#9999#*****-*********************,****--AAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAA.....
       354   /.*****-#999#******---****--*****************.////////**A-AAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAA.....
       356   ,*8999-*******-*********--********..////////////////,***--AAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAA.../*/
       358   #...**--**-*******-*****--**.///////////////////,**A-AAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAA-AAAAAA/////
       360   /..*,**--******--*******---***/////////////////////***-AAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAA*AAAAAA/*///
       362   ...*,.*--*******--********.//////////////////////*-*-AAAAAAAAAAAAA-AAAAAAAAA,AAAAAAAAAAAAAAAAAAAAA.AAAAAA/////
       364   ....,**-*********-*********..//////////////////,***--AAAAAAAAAAAAAAAAAAAAAAAAA./AAAAAAAAAAAAAA.AAAAAAAAAAAAAA/////
       366   ..*.,*-*********-**********./,/////////////////,**A-AAAAAAAAAAAAAAAAAAAAAAA,AAAAAAAAAA-AAAAAAAA*AAAAAAAAA/////
       368   ..****--*********--*--*****//.//////////////////***-AAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAA/////
       370   ..****--*---**-*---*****...///////////////////.-.-AAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAA*AAAAA*AAAAAAAAAA/////
       372   ...*-*********-*----****-***/,////////////////,**-AAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAA.AAAAAAA,AAAAAA/////
       374   ....,.******----******...//////////////////,**.AAAAAAAAAAAAAAA.AAAAAAAAAAAAAAAAAAAAAAAAA./,/AAA/AA/////
       376   ....********------*-*********/////////////////,***-*AAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAA/A./,AAAAAA/////
       378   /..***-------***-*********.//////////////////,***A*AAAAAAAAAAAAAAAAAA.AAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAA/////
       380   ...,.****---*-***-******,***//////////////////,****-AAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAA/////
       382   ...*******---*--*******,**///////////////////,**-AAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAA/////
       384   ....******----*-****---**///////////////////,**-AAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAA/////
       386   ,.*--*-*---*--*-******--*.///////////////////,***-*AAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAA/////
       388   *******---*-*-****-*******.///////////////////,**-*AAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAA/////
       390   ..******----***-**********.////////////////////.*****AAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAA//////
       392   ***-**-----****-----**--****.//////////////////,**A*AAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAA//////
       394   ***-*-*-##--*-------****-*****.///////////////***-*AAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAA/////////
       396   ***----,--#-----------*****-***.////////////////,**-*AAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAAA//...,/..//////////
       398   ****-----------------*****,***.//////////////////,**A-AAAAAAAAAAAAAAAAAAAAAA,,,..////...//.////...//////////
       400   .****---------------*************./////////////..**.*******AAAAAA/.........../.......///////./////////...//////////////


       POINTS PER CLUSTER IN THIS FIELD
        CLUSTER      SYMBOL       POINTS

            1           /           653
            2           *           691
            3           .           502
            4           ,             4
            5           A           236
            6           #            19
            7           -           257
```

ORIGINAL PAGE IS
OF POOR QUALITY

```
FLD 2

TOTAL NUMBER OF POINTS IN THIS FIELD   2860
LINE  644 THROUGH  694 EVERY  2 LINE(S)   SAMPLE   1 THROUGH 220 EVERY  2 SAMPLE(S)


       00000000000000000000000000000000000000000000000000111111111111111111111111111111111111111111111111112222222222
       00000111112222233333444445555566666777778888899999000001111122222333334444455555666667777788888999990000011111
       13579135791357913579135791357913579135791357913579135791357913579135791357913579135791357913579135791357913579

  644  /....*****.****.*.*...*..*/./.........../....*..Δ*.Δ---A....A.X-***********--*.---*----**--------------**---*ΔΔΔΔΔ.//
  646  Δ+++++++++BB*++BBB9BBB+.*************-*-**-*-***-****.,*********.X-**********--*.---------*--------****-----+ΔΔΔΔΔ///
  648  ++++B++++B++++++BBB8B+B*****-*****-*-*-***-*-*****-*************-.X***-*******--*-------------*****---X--+ΔΔΔΔΔA//
  650  Δ+++++++++BB+++BBBBBBBB*..-*.******--*----*----*************-.X-*--*-***----*---------*-----------****------+ΔΔΔΔΔ.//
  652  Δ.+++++++B++B++BBB+++B+B*******-----------------------**.****.*****,B-**-----------------*----------------++ΔΔΔΔΔA//
  654  ΔΔΔ+++BBBBBBBBBB+BBBBΔ*.*-*---*-----------------*************.X-**---X--------------------------------ΔΔΔΔΔA//
  656  ΔA+Δ+++B+++BBBB++++BB9-------*---------------------**-********-.B----------------------*------------**-++ΔΔΔA//
  658  Δ++++++BBBBBBB+++++BBB+--------*---------------*-************--*-,X---*--------------------**-----*--****---+ΔΔΔΔΔ//
  660  +++++BBBBBB+++++++BBBB+*-*---**-*------*-*------*-**-.***********-ΔX***-----------------.*---*-----------*******ΔΔΔΔΔA//
  662  ++++++B+++B++BEBBBBBB+---****------------------**************ΔX**,****---**-------*-----------------*********ΔΔΔΔΔA//
  664  +++++++++++++++BBBB++BB+**************-**---**-*-------****************-,X-*********-------*,*******************.ΔΔΔΔΔA//
  666  B++++++++++++++B++++++++***---*****-**---*-**-----***************-,X-*********----------**********.....,......../ΔΔΔΔΔΔA//
  668  B++++++++++++++++++++++++***-**----------*---------*************.-AX-***.****--*..,/.//..,/........//......./../ΔΔΔΔΔA//
  670  ++Δ++++++++++++++++BBBB+**********------*-----------***************-*X.........//////////...//.........///.//../ΔΔΔΔΔA//
  672  +++++++++++++Δ++++BB+B+B-----****--------------------**-*************.*X./////..///////////////..////////////./////////ΔΔΔΔΔA//
  674  ++++++++++Δ+ΔΔΔ+++++++++BΔ*..-*--*------------------*.-***********.ΔX....././/////////////////////////////////.ΔΔΔΔΔΔΔ//
  676  ++++++Δ+++++++++BB+BBBBB+*--*-*-----------------------****.**.*.X-../////////////////////////////////////ΔΔΔΔΔΔ//
  678  +++++++++++++++++BBB+++BB+*------------------*.*-------*****...-.X*..//././/////////////////////////////////ΔΔΔΔΔA//
  680  ++++++++++ΔΔ++++++++++B+*-*-**-------*-----*-*--------*******/./*/X..////.//////////////////////////////ΔΔΔΔΔA//
  682  +Δ+++++++Δ+Δ++++++++++++++++----------.*-*-*--*---*****+-*/*-.......*////////////////////////////ΔΔΔΔΔΔ//
  684  ΔΔΔΔ++++++++++++++++++B++.-------------------.-**---*.*,++/X-...///.////////////////////////////ΔΔΔΔΔΔ./
  686  ΔΔΔΔ+++++++++BB+++BBBB+*-------------------.*-***---*...AX../...//////////////////////////////ΔΔ+ΔΔΔ./
  688  ΔΔΔΔ++++BBBBBBBBBBBBBB8+--------------------*.-**-/9*-X--*-ΔX...../////////////////////////////ΔΔΔΔΔΔ//
  690  ΔΔΔA++++BBBBBBBBBBB+B+BA+-----------------***-****.-*,*-*X....//////////////////////////////ΔΔΔΔΔA//
  692  ΔΔ++++++BBB+B+BBBBBB+B++Δ-----*-------------*******-+-*---*B-...//////////////////////////////ΔΔΔΔΔA/.
  694  ΔΔΔ++++BB+BB++BBBBBB+BB+----**--------------.***-***-*******B-./////////////////////////////ΔΔΔΔΔ./.


POINTS PER CLUSTER IN THIS FIELD
 CLUSTER       SYMBOL       POINTS

     1            /           516
     2            *           599
     3            .           166
     4            +           378
     5            Δ           194
     6            X            25
     7            -           739
     8            B           167
     9            9             3
    10            A            29
```

ORIGINAL PAGE IS
OF POOR QUALITY

6.0 References

1. Ball, G. H., and Hall, D. J.; A Clustering Technique for Summarizing Multivariate Data, Behavioral Science, Volume 12, 1967.

2. Ball, G. H., and Hall, D. J.; ISODATA, A Novel Technique for Data Analysis, and Pattern Classification. Technical Report, Menlo Park, California, Stanford Research Institute, May 1965.

3. Kan, E. P. F.; "The Latest Version of ISODATA(A)/ISOCLS", Lockheed Electronics Company, Inc., HASD, Houston, Texas, Tech. Memorandum TM642-570, Sept. 1972.

4. Kan, E. P. F. and Holley, W. A.; "More on Clustering Techniques with Final Recommendations on ISODATA," Lockheed Electronics Co., Inc., HASD, Houston, Texas, Tech. Rep. 640-TR-112, May, 1972.

5. Kan, E. P. F., "Data Clustering: An Overview, "Lockheed Electronics Co., Inc., HASD, Houston, Texas, Tech. Rep. 640-TR-080, Mar. 1972.

6. Kan, E. P. F., "Adaptive Training Class Statistics," Lockheed Electronics Co., Inc., HASD, Houston, Texas, Tech. Rep. 640-TR-073, Feb 1972.

7. Earth Resources Data Format Control Book, Vol. 1, PHO-TR543.

APPENDIX A

Appendix A is an alphabetical list of the diagnostic messages printed out by the program.

1. A LINE NO. IS LESS THAN OR EQUAL ZERO.

This message is printed by subroutine TAPERD. Each data record on the input data tape contains the scan line number. After unpacking these bits a test is made to see if the number is greater than zero, if not the above message is printed and execution terminates.

User should check to see if correct data tape has been assigned to FORTRAN unit 3 (Logical unit C), and that the input card for FORMAT corresponds to the format of the data tape.

2. AT LEAST ONE FEATURE MUST BE INPUT.

This message is printed by subroutine SETUP. The user has failed to input the channels he wishes to use in the analysis. See FEATURES input card in section 3.1.1.

3. AT LEAST ONE FIELD MUST BE INPUT

This message is printed by subroutine SETUP. The $END* card was encountered before any fields were defined. The user must define at least one field by lines and columns. See section 3.1.1.

4. ERROR READING FIRST DATA RECORD--ISTAT= ————.

Printed by subroutine TAPERD. See Explanation of 6.

5. END-OF-TAPE ON UNIT —— LAST LINE WRITTEN.

Printed by DASTAP.

6. ERROR WHILE READING DATA RECORD.

LAST SCAN LINE READ —— ISTAT= ————.

Printed by subroutine TAPERD. Either a parity error (ISTAT= -3) has been encountered on the data tape or a device error (ISTAT= -4) has occurred. Resubmit the run with a back-up tape if available. Before resubmitting make sure you have checked the list of possible errors printed with this diagnostic.

7. FIELD BOUNDARY FOR FIELD —— DEFINED BEYOND SCOPE OF DATA. THIS FLIGHT LINE CONTAINS —— SCAN LINES.

Printed by subroutine TAPERD. The last scan line on the data tape is less than the last line of one of the fields defined by user. The last line on the tape becomes the last line in the field and the program continues.

8. FILE ASSIGNMENT NOT MADE FOR UNIT (4 or 7) NSTAT=——.

Printed by subroutine SETUP. An attempt was made to assign a word length to drum unit 4 or 7 which was more than the number of words available.

9. IN FIELD —— INITIAL (SAMPLE OR LINE) IS GREATER THAN LAST (SAMPLE OR LINE).

Printed by subroutine SETUP. An error was made in the definition of one of the fields, check input card.

10. INITIAL (SAMPLE OR LINE) IN FIELD —— DEFINED LESS THAN 1.

Printed by subroutine SETUP. Check input card for indicated field.

11. INTERNAL DIMENSIONS TOO SMALL FOR DATA.
NO. OF CHANNELS ON DATA TAPE = —— NO. OF POINTS/ CHANNEL = ——.

Printed by subroutine TAPERD. The most data the array IDAT, which is dimensioned by the parameter variable MAXDIM, must contain at one time is (1) the packed scan line as read from tape (2) the unpacked scan line and (3) at least one data point to be clustered. The storage needed for the above data is calculated and if it is greater than MAXDIM the diagnostic is written and execution terminates. The number of channels and points per channel as unpacked from the header record of the tape are written out with the diagnostic. If these numbers are in error then the header record of the tape is bad and must be corrected.

12. INVALID INPUT CARD -- IGNORED

Printed by subroutine SETUP. Check section 3.1.1 for correct spelling of key-words for card input and make sure the key word is left-justified in the field.

13. $\left(\begin{array}{c}\text{LINE}\\ \text{or}\\ \text{SAMPLE}\end{array}\right)$ INCREMENT IN FIELD —— IS LESS THAN 1.

Printed by subroutine SETUP. Check input card for indicated field.

14. SCAN LINES ARE NOT SEQUENCED CORRECTLY AROUND LINE ———.

Printed by subroutine TAPERD. Scan lines on data tape are not sequenced by 1. An attempt is made to continue reading next field.

15. TOO MANY FIELDS ARE BEING INPUT.

Printed by subroutine SETUP. The number of fields is limited to the parameter variable MAXFLD, which is set to 100.

16. UNRECOVERABLE ERROR READING HEADER RECORD.

Printed by subroutine TAPERD. A parity error or a device error was encountered in attempting to read the header record from the data tape. This usually indicates the tape is bad. See explanation of diagnostic 6.

17. USER HAS REQUESTED TOO MANY DATA POINTS---STORAGE
NOT AVAILABLE. AVAILABLE STORAGE ON FH432 DRUMS
IS ———WORDS. AVAILABLE STORAGE ON FH1782 DRUMS
IS ———WORDS.

Printed by subroutine SETUP. The number of data points times the number of channels requested by user exceeds the storage available on FH432 drum, or the number of data points exceeds the storage available on FH1782 drum. User must reduce the amount of data to be clustered.

18. WRITE ON UNIT ——— TERMINATED ABNORMALLY. DAS TAPE
NOT CREATED. ISTAT= ———.

Printed by subroutine DASTAP. User should resubmit the job with a different output tape. This diagnostic indicates that either a bad type was being used or a tape drive error occurred.

APPENDIX B

Aircraft Data Storage Tape Format

LARSYS Version 2

There are four types of (physical) records on the Aircraft Data Storage Tapes. They are:

1. ID record - 200 full words fixed length
2. Data record - variable length
3. End-of-Tape records - 200 full words fixed length
4. End-of-File records - IBM Standard

An aircraft Data Storage Tape contains one or more data runs consisting of an ID record, several data records and an End-of-File record. After the last data run on the tape, an End-of-Tape record and two End-of-File records are written on the tape.

For the purposes of this presentation a word is defined to be 32 bits and a byte to be 8 bits. Further details regarding the physical records follows:

1. ID record (200 full words fixed length)

	Format	Description
ID(1)	Integer	LARS Tape Number (e.g., 1, 17, 102, etc.)
ID(2)	Integer	File number on this tape
ID(3)	Integer	Run number (8 digits aabbbbcc) aa - last 2 digits of the Year data was taken

	Format	Description
		bbbb - running serial number for the year data was taken cc - uniqueness digits for runs which would otherwise have the same run number
ID(4)	Integer	Continuation Code ID(4) = 0 means the first line of data follows this ID record ID(4) = X means that the data following this ID record is a continuation of a flight line started on tape X
ID(5)	Integer	Number of Data Channels (Spectral bands) on tape (30 maximum)
ID(6)	Integer	Number of Data Samples per channel per line
ID(7-10)	Alpha-numeric (4A4)	Flightline Identifications (16 characters)
ID(11)	Integer	Month data was taken
ID(12)	Integer	Day data was taken
ID(13)	Integer	Year data was taken
ID(14)	Alpha-numeric (1A4)	Time data was taken