"AS-BUILT" DESIGN SPECIFICATION OF THE CAMS/CAS INTERFACE TAPE REPORT GENERATION PROGRAM FOR LACIE 8

Job Order 71-593

(TIRFs 78-0016 & 78-0017)

(This document supersedes LEC-12022)

Prepared By

Lockheed Electronics Company, Inc.

Systems and Services Division

Houston, Texas

Contract NAS 9-15200

For

EARTH OBSERVATIONS DIVISION

SCIENCE AND APPLICATIONS DIRECTORATE

National Aeronautics and Space Administration

LYNDON B. JOHNSON SPACE CENTER

Houston, Texas

May 1978 LEC-12267

JSC-14238

"AS-BUILT" DESIGN SPECIFICATION OF THE CAMS/CAS INTERFACE TAPE REPORT GENERATION PROGRAM FOR LACIE 8

Job Order 71-593

(TIRFs 78-0016 & 78-0017)

PREPARED BY

K. P. Eckel

APPROVED BY

for James A Wilkinson

Philip L. Krumm, Acting Supervisor
Scientific Applications Section

Prepared By
Lockheed Electronics Company, Inc.
For
Earth Observations Division

NATIONAL AERONAUTICS AND SPACE ADMINISTRATION
LYNDON B. JOHNSON SPACE CENTER
HOUSTON, TEXAS

May 1978 LEC-12267

CONTENTS

Section Page

1. SCOPE

1.1 GENERAL

This document is the "as-built" design specification of the CAMS/CAS Interface Tape Report Generation Program for LACIE 7.

2. APPLICABLE DOCUMENTS

- TIRF 77-0052
- TIRF 77-0040
- Specification for the CAMS/CAS Interface Tape Report Generation Program - LEC-9151
- CAMS/CAS Interface Control Tape format specification in Earth Resources Data Fromat Control Book Volume 1 (PHO-TR543, Rec. A. Change 3)
- TIRF 78-0010
- CAMS/CAS Interface Tape Printout after LACIE 6A JSC Memorandum, SF4-77-7-13, 7/21/77.
- "As-Built" Design Specification of the CAMS/CAS interface tape Report Generation Program LEC-11292
- Acceptance Test Specification For CAMS/CAS Interface tape report generation Program for LACIE 7-LEC-11787
- Clafification and Prioritization of LACIE 7 CCIT Report JSC memorandum, SF4-77-11-8, 11/4/77.
- TIRF 78-0016, Intermediate Values of Discriminate Function, Feb 78.
- TIRF 78-0017, CAMRPT Subclass Category Expansion, Feb 78.
- Acceptance Test Specification for subclass category expansion and Intermediate value display values to CAMS/CAS Interface Tape Report Program. LEC-12164.
- TIRF 78-0026, Combined Categories
- Acceptance Test Specification for the Combined Categories change to the CAMS/CAS Interface Tape Report Program. LEC-12613

3. SYSTEM DESCRIPTION

3.1 HARDWARE DESCRIPTION

N/A

3.2 SOFTWARE DESCRIPTION

The purpose of this program is to produce CAMS reports from data on the CAMS/CAS interface tape.

3.2.1 SOFTWARE COMPONENT NO. 1 (CAMRPT)

The main program CAMRPT reads control cards, locates segment data on the input tape and calls subroutines to generate requested reports.

3.2.1.1 Linkages

CAMRPT calls subroutines CDRED, CAMREC, BAUEXT, CLURES, STDATA, CONRED, and DOTRAY.

3.2.1.2 Interfaces

N/A

3.2.1.3 Inputs

CAMRPT control cards are: SEGMENT XXXX, RECORD ID XXXXXX XXXXXX, ALL, END. CAMS/CAS interface tape records are inputs to CAMRPT. See reference 4 in section 2, for record formats.

3.2.1.4 Outputs

An error message is output indicating a bad data card. If a requested segment is not on the input tape, the program writes a message to that effect.

3.2.1.5 Storage Requirements

Total space allocated is 8474 bytes.

3.2.1.6 Description

CAMRPT is the CAMRPT main program. The program sets the printout option indicator PRTOUT to 0 initially. In this mode the output of some reports is conditional. On the first call to tape read, subroutine CDRED, the program reads two data cards specifying the device code (M or X) and the unit number (0 to 1), then calls CONRED to read constants to be used in the calculations for DW & DS and the subclass categories to be combined. CAMPRT next reads a program control card and tests the first non-blank character for one of the following: S, R, A, or E. If the card is blank or the first character is not one of the above, the program prints an error message on the line printer and stops.

The action taken for each control card is given below. Note that if a control card other than E is read in, the printout option indicator PRTOUT is set to 1. This is the option to output all reports, including conditional reports.

- S - Option indicator PRTOUT is set to 1. The program obtains the segment number from the input card. The program searches the input tape for a recognition segment record whose segment number matches the control card segment number. If a match is not found, a message is printed and the program goes to read the next control card. If a match is found, the reports for the segment requested are generated. The program then reads the next control card.

  program goes to read the next control card. If a match is found, the reports for the segment requested are generated. The program then reads the next control card.

- R - The action taken is the same as for the S card above, except that the record identification number is used instead of the segment number.

- A - Option indicator PRTOUT is set to 1. Beginning with the segment on the tape at which the tape is currently positioned, the program generates reports for that

segment and all the following segments. When the second tape end of file, indicating end of data, is reached, the program rewinds the tape and returns to read the next control card.

- E - If the printout option indicator PRTOUT is 0, the program generates reports in the limited printout mode, rewinds the tape, and then stops. If PRTOUT is 1, the tape is rewound and the program stops.

The tape read subroutine CDRED tests all records for valid characters. If any invalid data is encountered an error message is output to the line printer to inform the user that the data for that segment or record is questionable. Any invalid characters in the record are converted to ones (1) and normal processing is resumed.

To generate reports for a segment the program first calls CAMREC to process recognition segment records. The processing entails generation of the optional classification Data report, the standard Separability Report and the standard header sheet for the report.

After CAMREC, the programming calls BAUEXT to process the Bias correction results records. Data from the Bias correction results records is saved in common blocks BIAS and Dummy. CAMRPT next calls CLURES to process the clustering results records and to generate the conditional cluster report. In addition CLURES saves cluster dot data in common block CLDOT for later use in the optional Dot report. Next STDATA is called to process the statistics records and generate the optional Statistics Report. Finally DOTRAY is called to process the Dot subset records and generate the standard Dot Label/Classification, Bias Correction Classification, Dot Label/Cluster and Bias Correction Cluster Reports.

In the limited printout mode the decision as to whether or not to output the optional reports is not made until the percentage of

correctly classified Dots (PCC-1 & PCC-2 for Bias Correction Classification report) or percentage of correctly clustered Dots (PCC-1 & PCC-2 for Bias Correction Cluster Report) are calculated in subroutine DOTRAY. If any of the values are less than 80%, PRTOUT is set to 2 in DOTRAY and the conditional reports are read from disc and written to the line printer. In the full printout mode (PRTOUT=1) the conditional reports are always retrieved from disc and printed out. If PRTOUT was = 2 it is reset to 0 after the report for a segment has been output.

3.2.1.7 Flowcharts

See Flow Diagram 1.

3.2.1.8 Listing

See Appendix A.

3.2.2 SOFTWARE COMPONENT NO. 2 (CAMREC)

This program processess classification results contained in recognition segment records, outputs the conditional Classification Data report, saves data from the recognition segment records for use in generating the standard Separability Report and the standard CAMS Interface Report Header sheet.

3.2.2.1 Linkages

CAMREC is called by CAMPRT and calls subroutines BIAPRT, CPIPO, MV, CDRED, BNT and SEPRPT.

3.2.2.2 Interfaces

N/A

3.2.2.3 Inputs

Recognition segment records, containing subclass a priori and threshold values, subclass related classification results, and feature selection Bhattacharyya separability data for available acquisitions.

3.2.2.4 Outputs

CAMS Interface Report Header and a Classification Data report.

3.2.2.5 Storage Requirements

Total space allocated is 7397 bytes.

3.2.2.6 Description

CAMREC is called with the first recognition segment record for the segment to be processed residing in array IBUF. CAMREC first calls BIAPRT with PASS=1 to have the report heading, segment

number, record ID, and acquisition dates output. Next CAMREC saves the number of channels used in classification and the Bhattacharyya separability data from the first recognition segment record for later use in generation of the Separability Report.

Title and column headings for the classification section of the report are written out by CAMREC. Processing of classification results begins by setting the location in array IBUF of the first subfield containing subclass related results. Subfield contents are accessed by calling CPIPO. CPIPO returns the class portion of the subclass name and the counts PI and PO of pixels classified into, and thresholded out of the subclass. If the first character of the class name is X, PI is added to the X category pixel count. If the category is W, for wheat, then the count for the first wheat class is set to PI and the wheat class name is saved in CLIST. PO is added to the total of pixels threshold, TC, in the COMMON blocks CBIAS.

In processing for the second, and subsequent subclasses, the program calls CPIPO to get the next class name, checks to see if it is wheat, and, if so, compares it to the last class name in CLIST. If it is not the same, the new name is saved in CLIST and the class index is incremented by 1. This causes wheat class pixel count PI to be tallied in the next results array location.

After all classification data has been processed the feature selection Bhattacharyya separability data is saved from the last recognition segment record and SEPRPT is called to generate the normal Separability Report.

3.2.2.7 Flowcharts

N/A

3.2.2.8 Listing

See Appendix A.

3.2.3 SOFTWARE COMPONENT NO. 3 (BAUEXT)

This program saves data contained in the clustering bias correction and classification bias correction results records.

3.2.3.1 Linkages

BAUEXT is called by CAMRPT. It calls CDRED.

3.2.3.2 Interfaces

N/A

3.2.3.3 Inputs

Clustering Bias correction and Classification Bias Correction results records.

3.2.3.4 Outputs

None.

3.2.3.5 Storage Requirements

Total space allocated is 6861 bytes.

3.2.3.6 Description

BAUEXT is called when the main program reads the first clustering Bias correction result record. BAUEXT saves the following data from both the Clustering Bias Correction and Classification Bias Correction results records for up to 8 categories of interest plus the "designated other" and "unclassified" category"

Pixel Population
Bias corrected estimator
Machine estimate
Random estimate
Variable of bias corrected estimate
Variance

In addition the number of categories of interest and the character used for the categories of interest are saved. All data is saved in common blocks CBIAS arrays. This data is used by CALC for certain calculations and by BIAPRT for output of the normal Bias Correction reports.

3.2.3.7 Flowcharts

N/A

3.2.3.8 Listing

See Appendix A.

3.2.4 SOFTWARE COMPONENT NO. 4 (CLURES)

This program processes the cluster results records and generates the conditional cluster report.

3.2.4.1 Linkages

CLURES is called by CAMRPT. It calls CDRED.

3.2.4.2 Interfaces

N/A

3.2.4.3 Inputs

Cluster results records.

3.2.4.4 Outputs

The conditional Cluster report.

3.2.4.5 Storage Requirements

Total space allocated is 10966 bytes.

3.2.4.6 Description

CLURES is called when the main routine reads the first cluster results record. The program decodes ALSETS, the total number of clusters, and SETSR, the number of clusters in the current record. The routine then outputs the cluster report header, ALSETS as clusters generated and any options used. Next CLURES saves all data for each cluster for later output. When all clusters in the current record have been processed, another cluster results record is read in and processed as above.

After all cluster have been processed and if the cluster/dot report option is set, four additional cluster results records need to be processed. The processing consists of saving all dot

information in an array called DOTBUF for later output. In addition each dots cluster assignment is transfered to the common blocks CLCOM for later use in a different report. Finally when all cluster results records are processed in the above manner the cluster information is output as follows. For each cluster the program outputs the cluster name, the Labeling dot match name, L1 distance, categorie used, brightness and greenness numbers for all Acquisitions used and information on all dots in the cluster. The clustering channel list is written at the end of the report.

3.2.4.7 Flowchart

See Flow Diagram 2.

3.2.4.8 Listing

See Appendix A.

3.2.5 SOFTWARE COMPONENT NO. 5 (STDATA)

This subroutine formats and outputs field and subclass statistics data.

3.2.5.1 Linkages

STDATA is called by CAMRPT. STDATA calls subroutines KNT, MDTTL, MEAN, POP, CDRED, SNAME, FANME, STDMP, and BNT.

3.2.5.2 Interfaces

N/A

3.2.5.3 Inputs

The statistics record, containing, for fields or for subclasses, the population and values of the mean and standard deviation by channel.

3.2.5.4 Outputs

The conditional statistics report.

3.2.5.5 Storage Requirements

Space allocated is 8360 bytes.

3.2.5.6 Description

STDATA is called from CAMRPT. By means of decode statements, the program converts several variables from input character format in IBUF to integers. The variables are ALSETS, the total number of statistics sets, SETSR, the number of sets in the current record, and NCH, the number of channels. STDATA calls 5 subroutines to move data from input record subfields to print buffers. SNAME and FNAME move name data and insert SUBCL and FIELD designations in the print buffer. POP is called to move

population data. MDTTL is called to supply column headings for means and standard deviations, which are transferred to a print buffer by MEAN. MEAN also puts decimal points where needed. The variable DSETS, set to 5, controls the number of statistics sets to be accumulated before outputting the print buffers. When the current record statistics sets counter reaches SETSR, and ALSETS sets have not yet been processed, STDATA calls CDRED to read the next statiscics record from tape. In addition, the pixel population and classified percentage for the category "unassigned" in the bias correction cluster report is calculated and saved in the common blocks CBIAS.

3.2.5.7 Flowcharts

N/A

3.2.5.8 Listing

See Appendix A.

3.2.6 SOFTWARE COMPONENT NUMBER 6 (DOTRAY)

This program processes Dot Data records and generates the conditional Dot Report.

3.2.6.1 Linkage

DOTRAY is called by CAMRPT. It calls BIAPRT, CALC and CDRED.

3.2.6.2 Interfaces

Dot Data records.

3.2.6.3 Inputs

N/A

3.2.6.4 Outputs

The Conditional Dot report.

3.2.6.5 Storage Requirements

Total space allocated is 6986 Bytes.

3.2.6.6 Description

DOTRAY is called by CAMPRT after reading the first Dot Data record. DOTRAY next outputs the report header to the top of the next page and processess the Dot Data records until all 209 dots are processed. DOTRAY also saves the dot Lable, dot type and classification for each dot in the common block CBIAS for later use by BIAPRT & CALC.

The processing involves outputting the following for each of 209 dots:

Dot number, line and pixel number for the dot, type and label (if any) for the dot, cluster and classification as well as the greenness and brightness of up to 4 acquisition for the Dot. Each record contains data for 15 dots. After the 15 dots in the record has been processed and output the next record is read in via CDRED and processed. After all dots are processed DOTRAY

calls CALC to perform calculation for the bias correction reports then calls BIAPRT to output the bias correction reports. Finally, DOTRAY check the value PRTOUT. If PRTOUT is 1 DOTRAY returns to the main program. If PRTOUT is not 1 all valid PCC values (percentages of TYPE 1 & TYPE 2 correctly classified or clustered dots) are tested. If any of the valid PCC's are less than 80%, then PRTOUT is set to 2, to indicate to the main program that the conditional reports are to be read from the disk and output to the line printer.

3.2.6.7 Flowcharts

N/A

3.2.6.8 Listing

See Appendix A

3.2.7 SOFTWARE COMPONENT 7 (SEPRPT)

This program processess the separability data which was saved for it by CAMREC and outputs the separability report.

3.2.7.1 Linkage

SEPRPT is called by CAMREC.

3.2.7.2 Interfaces

N/A

3.2.7.3 Inputs

Segment Recognition record number 1.

3.2.7.4 Outputs

The normal Separability report.

3.2.7.5 Storage Requirements

Total space allocated in 588 bytes.

3.2.7.6 Description

SEPRPT is called by CAMREC after all separability data has been saved. SEPRPT then outputs the separability report which consists of channel combinations for up 4 (16 channels) and selection Battacharyya separability data.

3.2.7.7 Flowcharts

N/A

3.2.7.8 Listings

See Appendix A.

3.2.8 SOFTWARE COMPONENT NO. 8 (CALC)

This subroutine calculates data necessary for the output of the Bias Correction Classification and Bias Correction Cluster reports.

3.2.8.1 Linkage

CALC is called by DOTRAY.

3.2.8.2 Interface

N/A

3.2.8.3 Inputs

Dot Data from common blocks CBIAS.

3.2.8.4 Output

None

3.2.8.5 Storage

Total space allocated is 9248 bytes.

3.2.8.6 Description

CALC performs 2 identical calculations on different sets of data. The first data set involves Classification data and the second involves Cluster data. CALC first sets all needed variables to zero. If the flag CLADUM is equal to 1, no classification calculations are performed and CALC goes directly to the cluster calculations. Otherwise CALC checks "category of interest" for the presence of an "N", or a match of a member of the subclass categories to be combined, and sets pointers and indication as necessary. Next certain arrays are zeroed.

Following are the variables calculated for each of 209 dots. It is understood, in every case, that the conditions apply to DOTS which

are labeled, that is, LBLED (I) is not blank, and to DOTS not classified as DU or DO.

- NTYP1 - The number of DOTS which are either type 1 or type 3.
- NTYP2 - The number of type 2 DOTS
- NAII - The number of type 1 and type 3 DOTS whose label and classification are the same.
- NGIJ - The number of type 1 and type 3 "GRAIN TYPE" DOTS whose label and classification are not the same.
- NOCL - The number of type 2 DOTS which are both labeled and classified.

In addition when calculating the above for all dots a bias correction vector table is calculated. This table consists of a two dimensional array and contains summations of dots that have valid labels as the first index and valid classification as the second index.

In addition the following variables are calculated: It is understood that classified does not mean "threshold."

- ALGT - The number of TYPE 2 dots labeled in a cateogry to be combined and classified in a category to be combined.
- ALBG - The number of TYPE 2 dots labeled with any "category used" and classified in a category to be combined.
- ALNT - The number of TYPE 2 dots labeled in any "category used" other than a category to be combined and classified in any "category used" other than a category to be combined.
- ALNB - The number TYPE 2 dots labeled in any "category used" and classified in any "category used" other than a category to be combined.

The subroutine computes the corrected percentages, uncorrected populations, variances, uncorrected percentages and random sample data for all "categories used" plus "combined category."

CALC now performs the identical calculations on the cluster data after checking CLUDUM as outlined above.

3.2.8.7 Flowchart

See Flow diagram No. 3.

3.2.8.8 Listing

See Appendix A.

3.2.9 SOFTWARE COMPONENT NO. 9 (BIAPRT)

This program outputs the report header sheet, Label/Classification table, Lable/Cluster table and the Bias Correction reports.

3.2.9.1 Linkage

BIAPRT is called by CAMREC & DOTRAY. It calls BNT.

3.2.9.2 Interface

N/A

3.2.9.3 Inputs

Segment recognition records and common blocks CBIAS, Dummy and CLCOM.

3.2.9.4 Outputs

Report header sheet, TYPE1 and TYPE2 Dot Label/Classification report, Bias Correction Classification Report, TYPE1 and TYPE2 Dot Label/Cluster report and the Bias Correction Cluster reports.

3.2.9.5 Storage Requirements

Total space allocated in 9965 bytes.

3.2.9.6 Description

When CAMREC calls BIAPRT it sets PASS=1. This causes BIAPRT to output the report header which contains tape number, DPAR No., record ID, segment number and all acquisition dates. This information is retrieved from the segment recognition record which had been read into IBUF.

When DOTRAY calls BIAPRT it sets PASS=2. In this mode of operation up to 2 similar reports can be generated. If CLADUM is equal to 1 no classification report is generated.

Otherwise DOTRAY generates an 11 by 19 matrix of user label/classification entries for type 1 and type 3 DOTS. A similar Matrix is generated for type 2 DOTS which also includes type 0 DOTS. DOTS with a classification label of DU or DO do not appear in either matrix.

Next the bias correction classification report is output using data calculated by CALC and stored in common blocks CBIAS. The report consists entries for all "categories of interest", DO, TH, UN and combined as follows:

Pixel population, classified and corrected percentages, variance and random sample estimate.

Also the alpha value matrix is output, and PCC values, DW and DS and lastly the Bias correction vectors and totals.

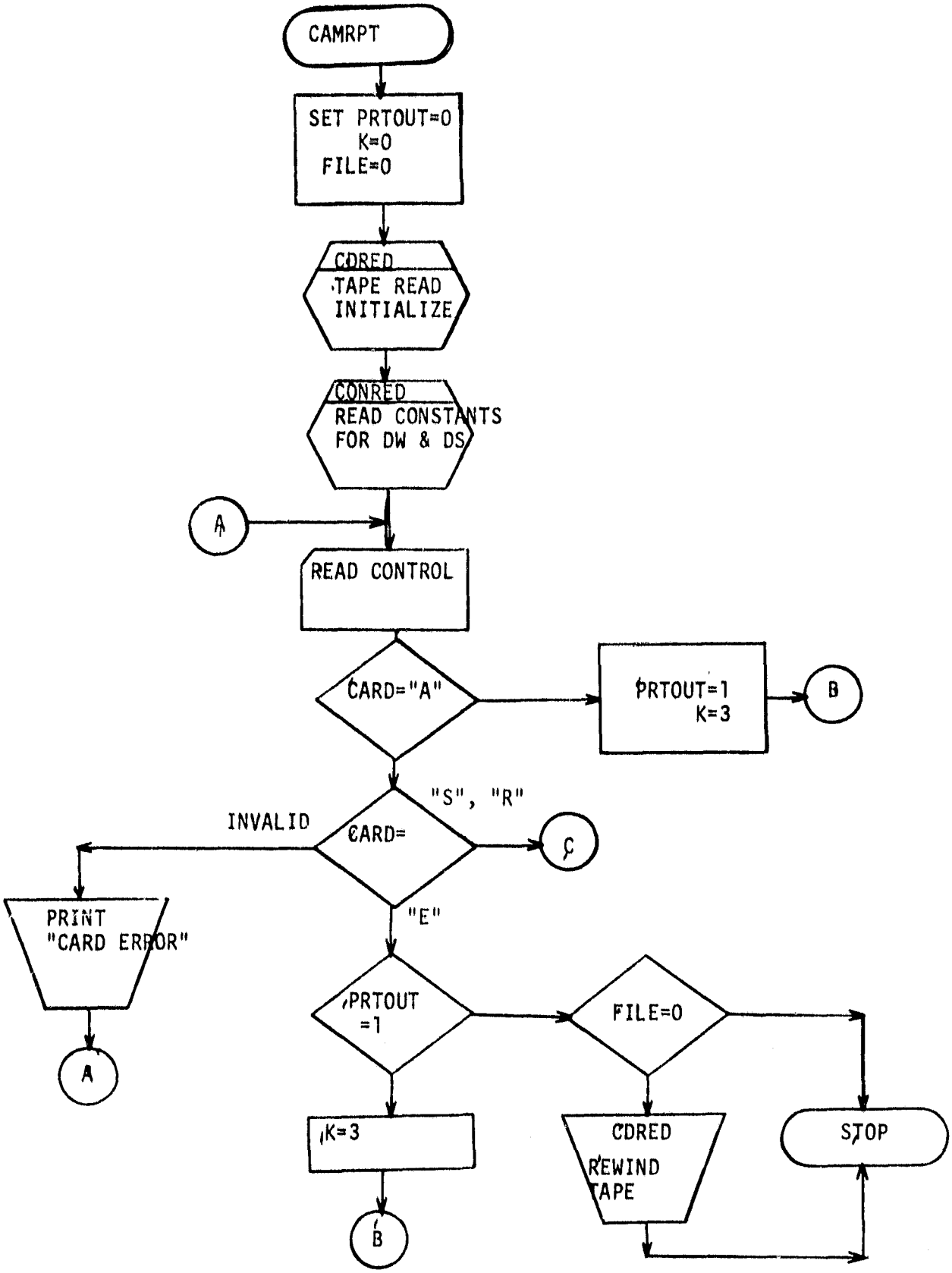
A similar report is generated for the cluster data if CLUDUM is not equal to 1.

3.2.9.7 Flowcharts

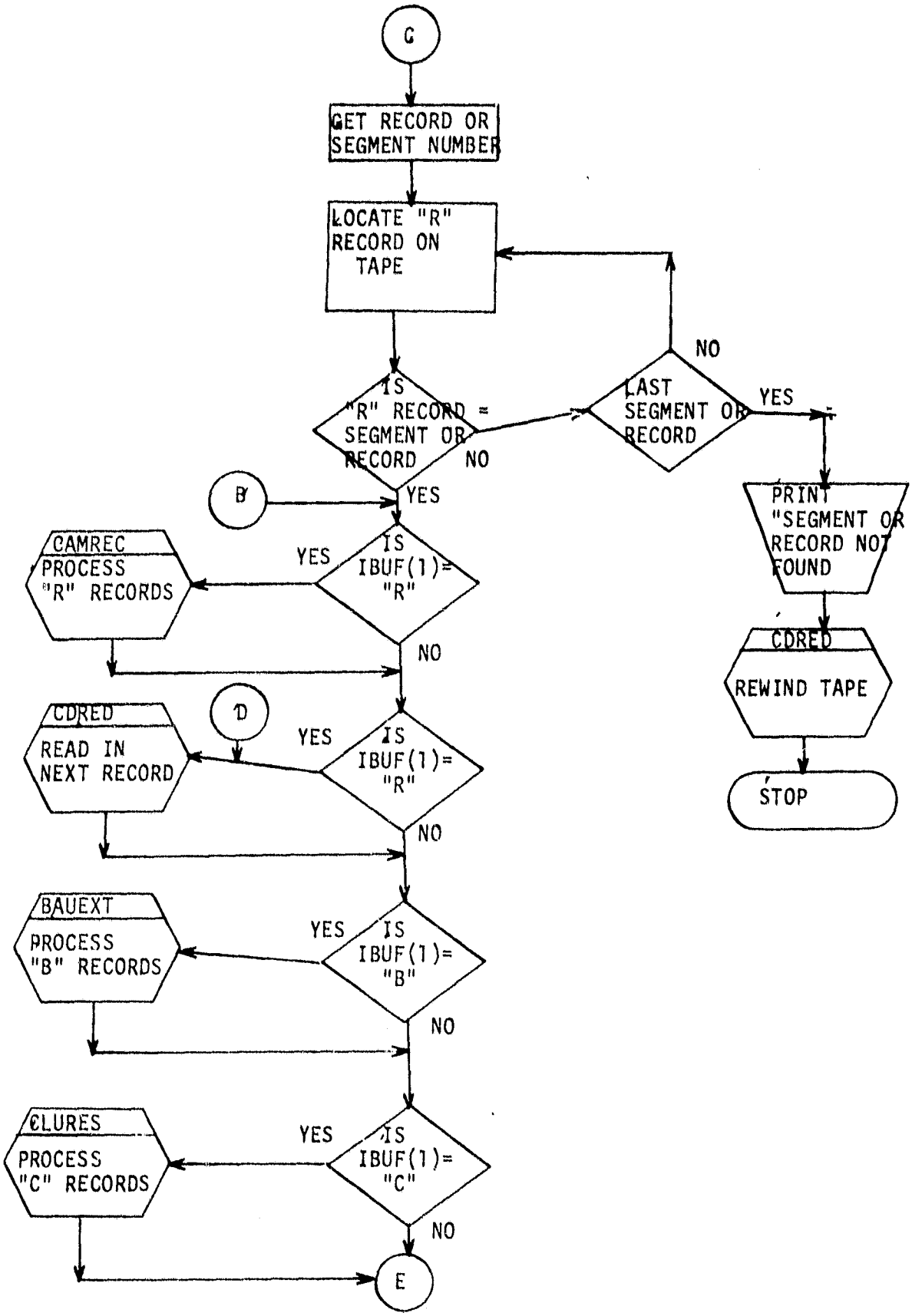
See flow diagram 4.

3.2.9.8 Listing

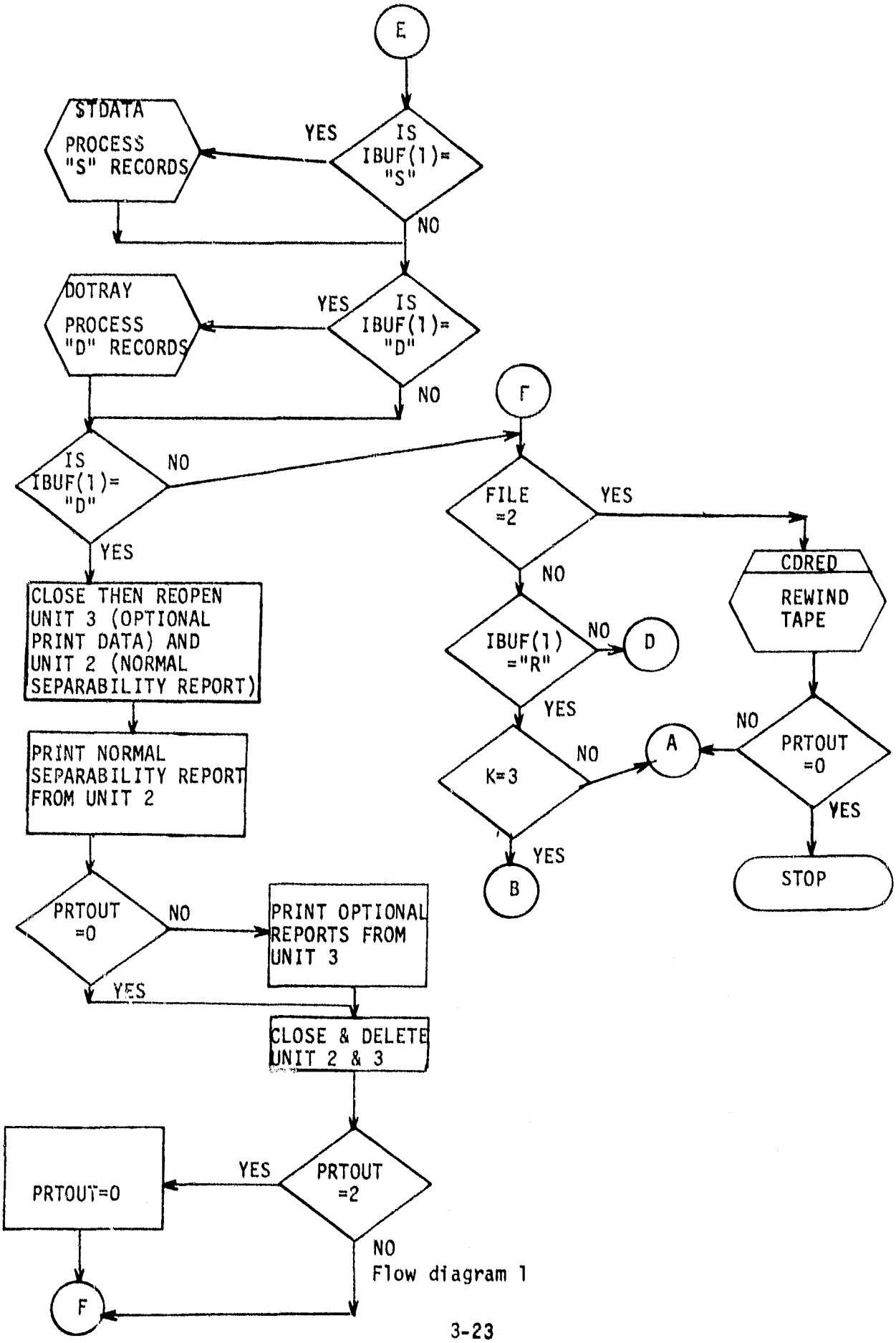
See Appendix A.



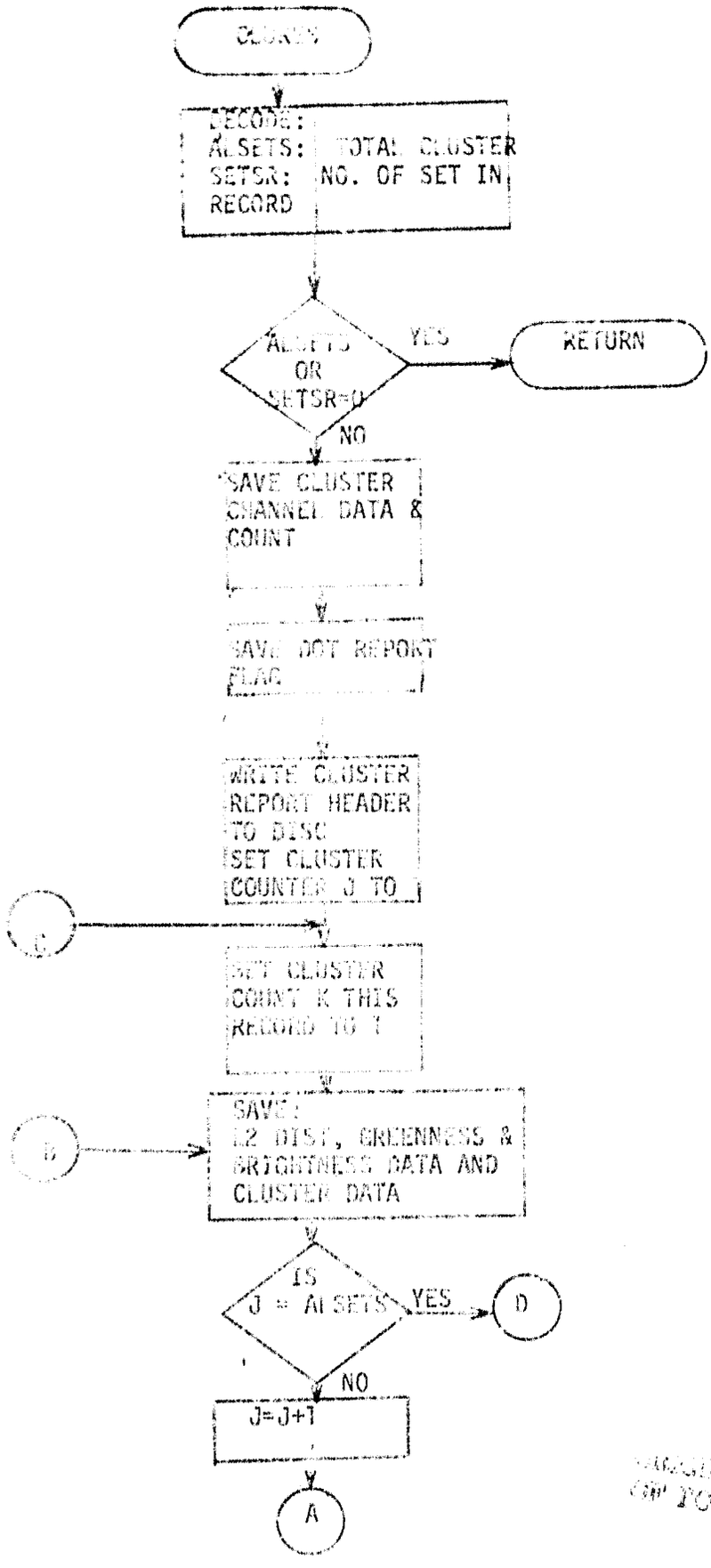
Flow diagram 1



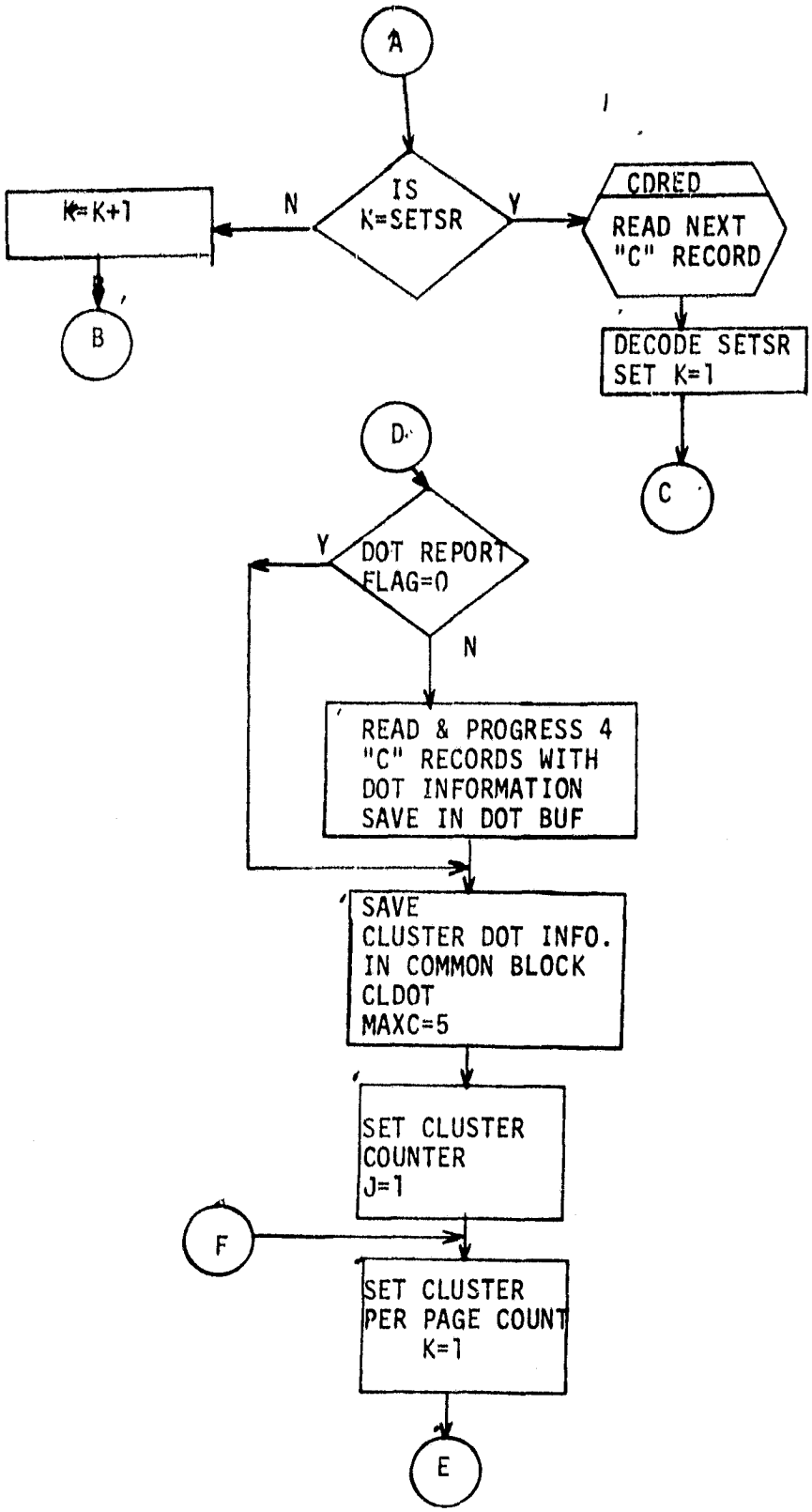
Flow diagram 1



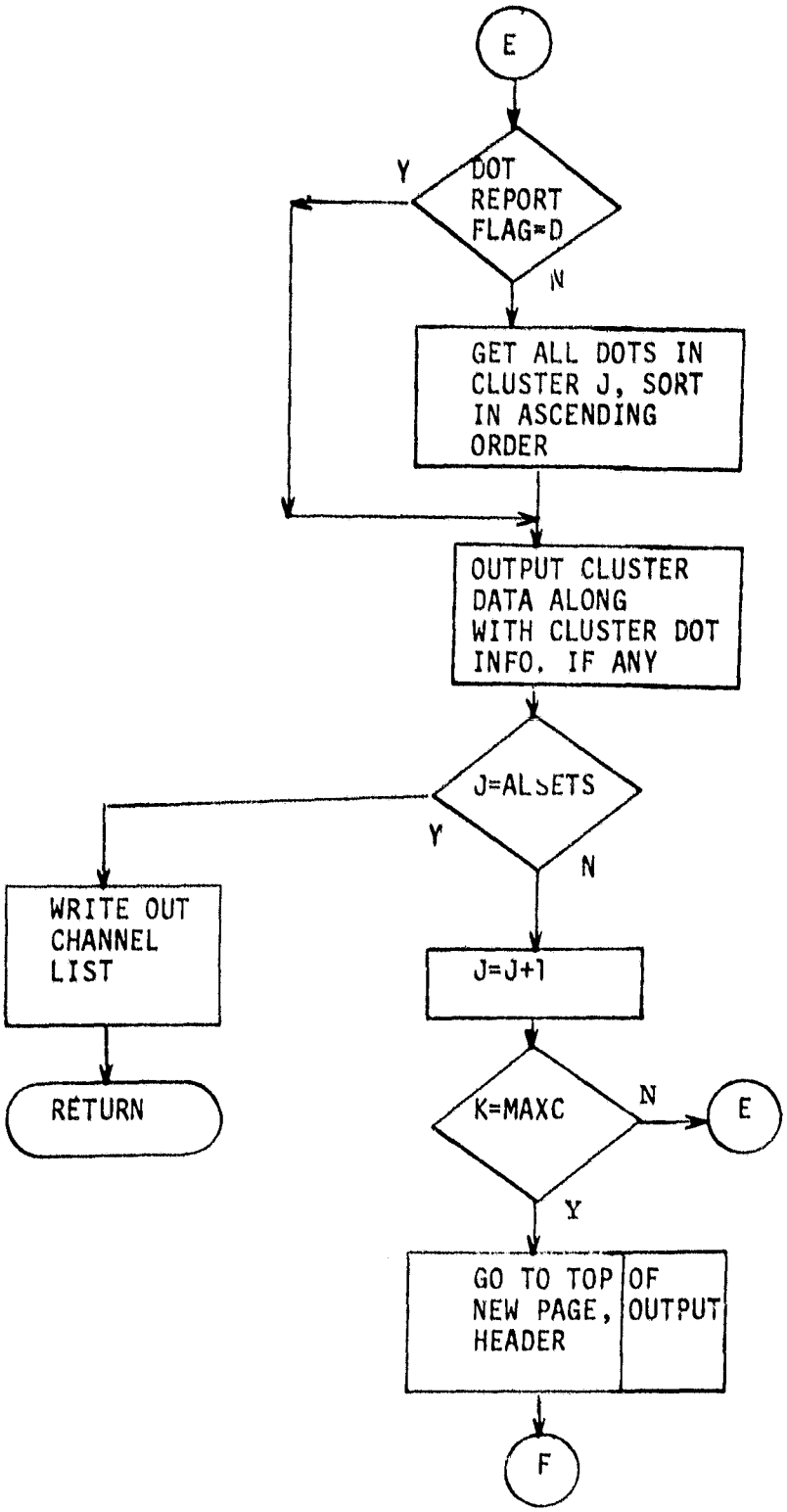
Flow diagram 1



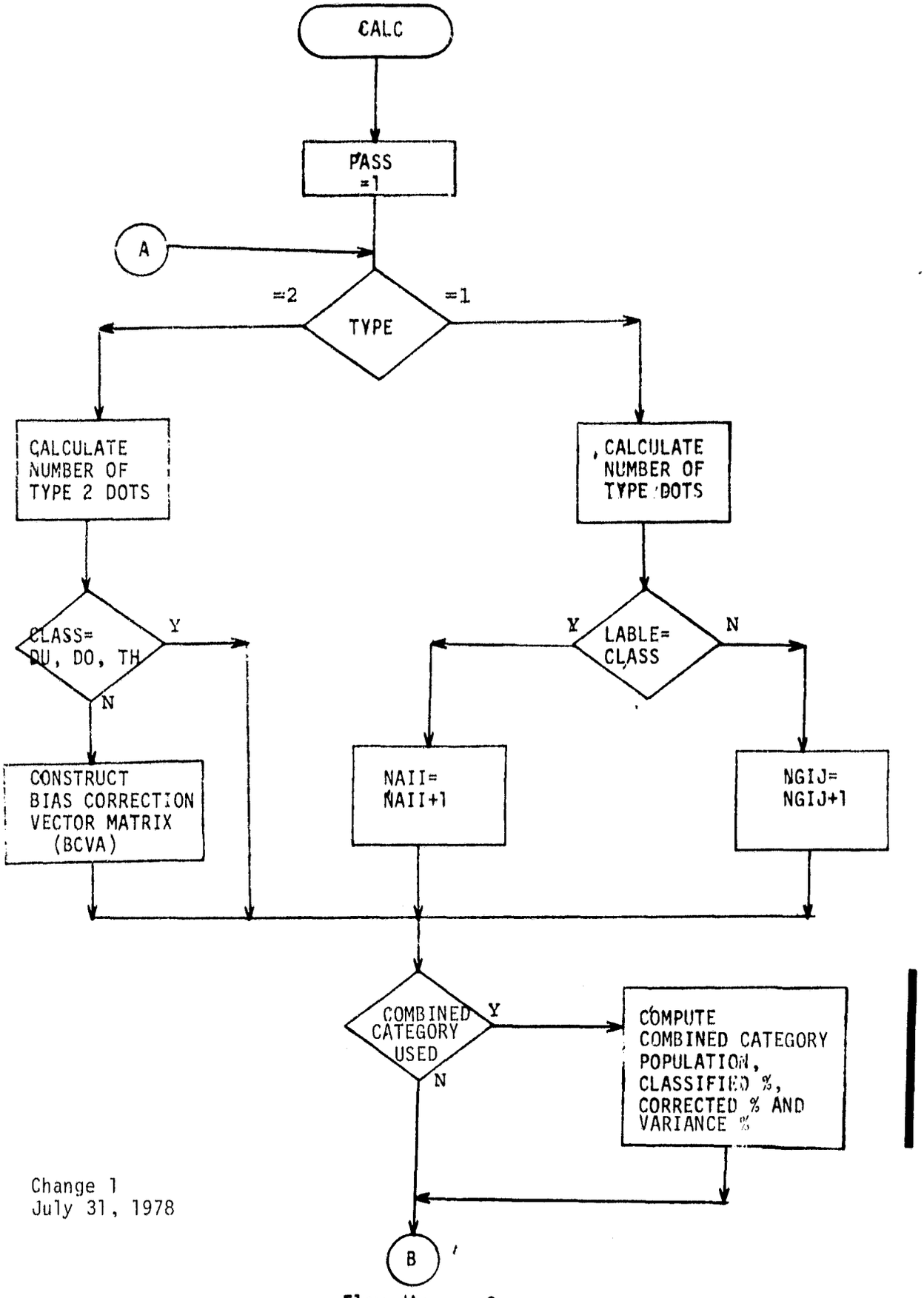
Flow diagram 2



Flow diagram 2

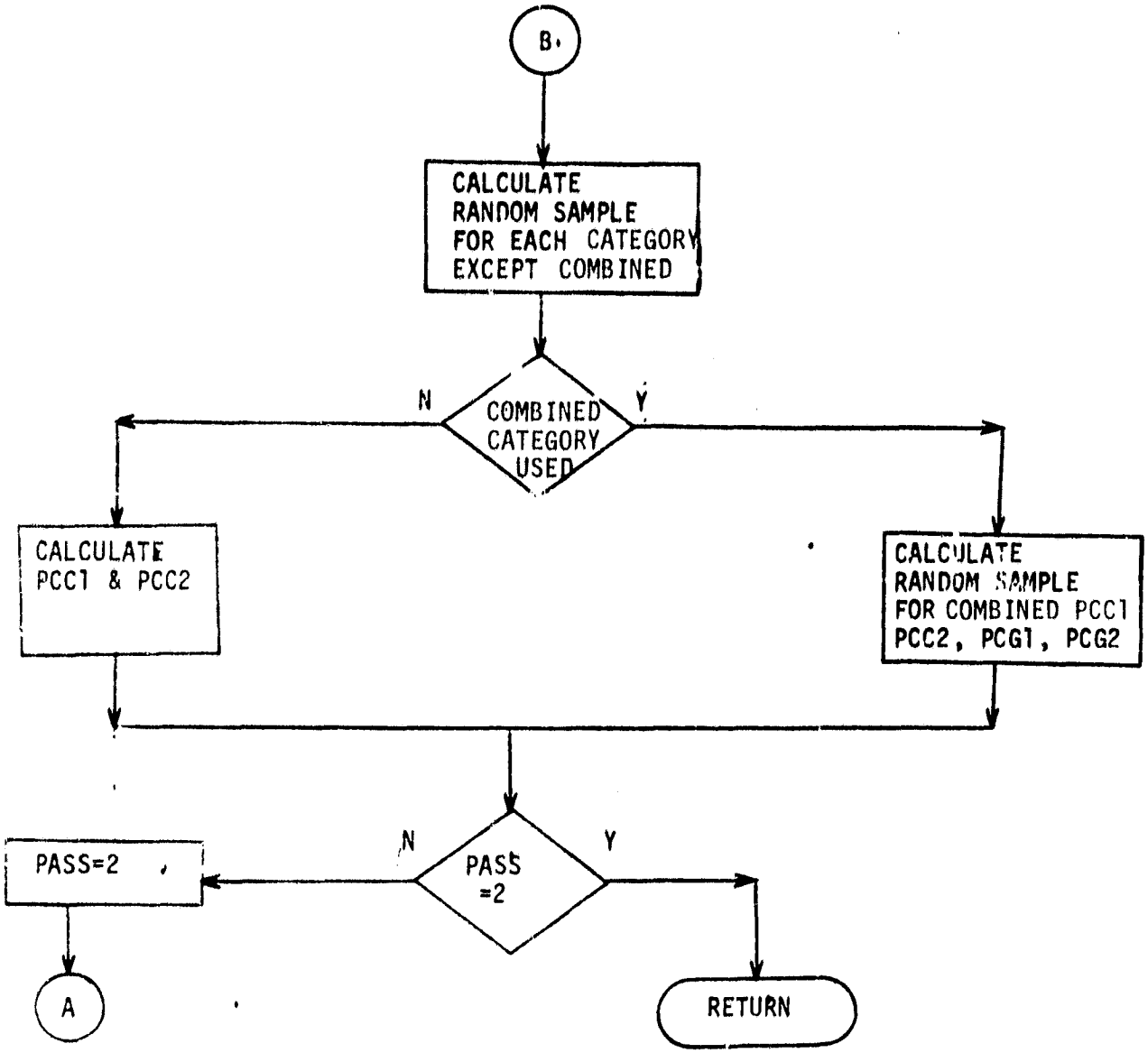


Flow diagram 2

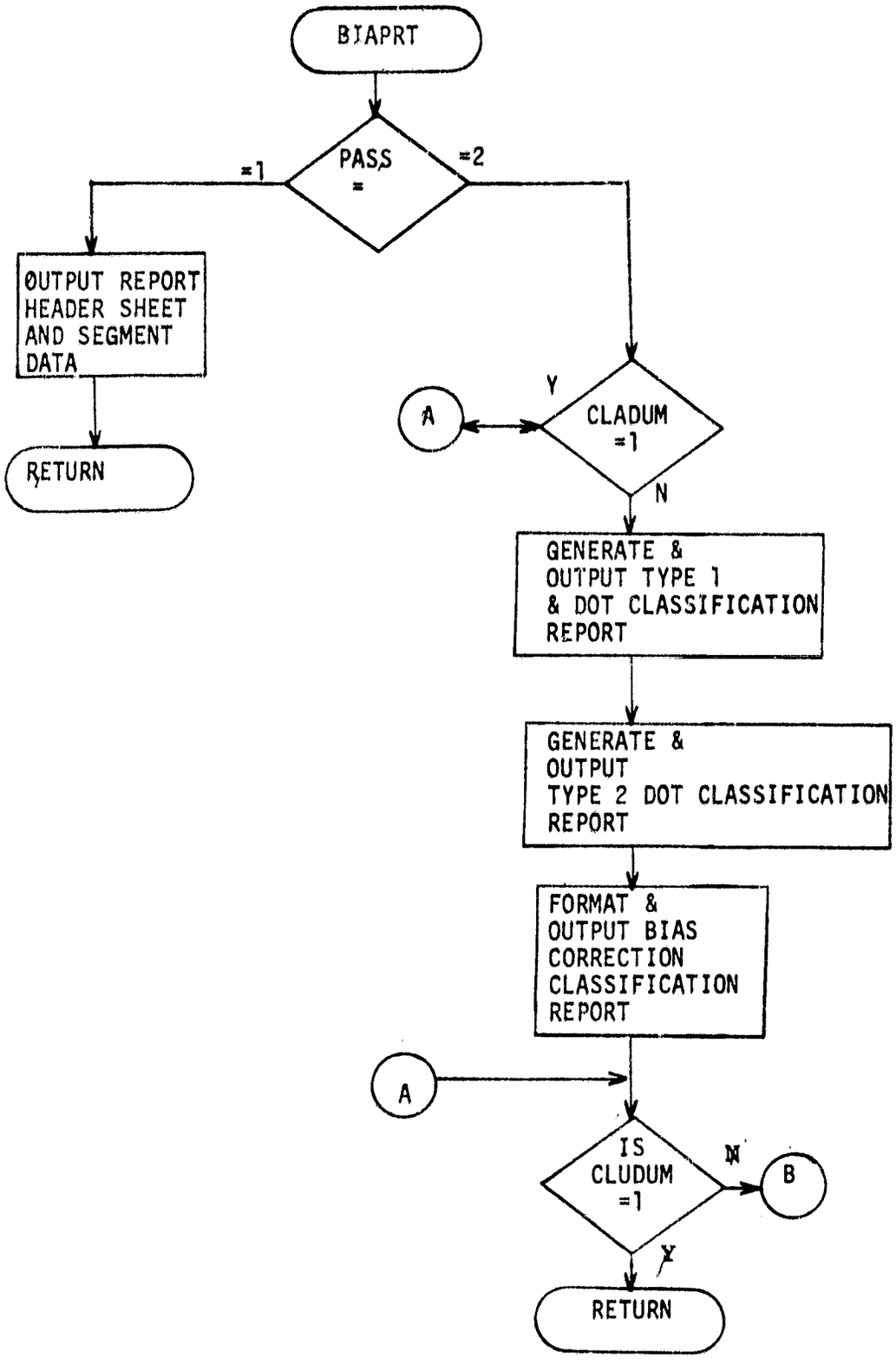


Change 1
July 31, 1978

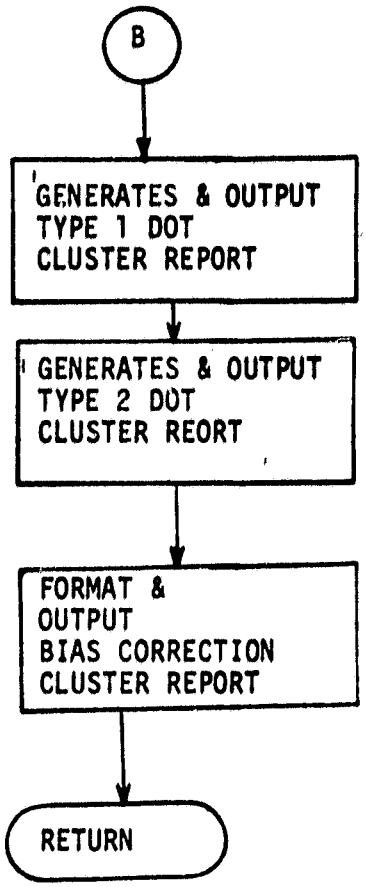
Flow diagram 3



THE LOGIC FOR SECOND PASS IS THE SAME AS THE FLOW CHART ONLY THE VARIABLES ARE CHANGED.



Flow diagram 4



Flow diagram 4

4. OPERATING PROCEDURE

4.1 GENERAL

CAMRPT is an RSX-11D Batch program which reads a CAMS/CAS Interface tape and generates a series of reports which are output on the line printer. It requires data card inputs.

4.2 DECK SET UP

The first 22 data cards define the input tape drive, input tape unit and curve fitting constants for DW & DS calculations. The DW & DS Formulae are shown in Appendix B. The card formats are:

M or X (tape drive)
0 or 1 (tape unit)
10 DW curve fitting constant cards as described in Appendix C
10 DS curve fitting constant cards as described in Appendix C
1 Combined category card as described in Appendix D.

Entries always start in column 1. To execute the CAMRPT default option for a limited printout of reports, an END card must follow the data cards above. If the option for a full output of all reports is desired, the control card sequence is:

```
A      (for all reports)
END
```

If the user desires to obtain the output for only a single segment on the input tape, segment 9681 for example, the control card sequence is:

```
S      9681
END
```

To obtain the output for segment 9681, and all segments following 9681, the control card sequence is:

```
S      9681
A
END
```

APPENDIX A

The above option is used when there is a bad segment on the input tape, to obtain the output for setments follcwing the bad seqment.

The Batch deck set up for the CAMRPT default option using input tape unit MTO is as follows:

```
$JOB/NAME=CAMRPT/MCR/LIMIT=99/ACCOUNT=50 50
$DATA
M
0
10 DW constant cards
10 DS constant cards
 1 combined category card
END
$EOD
$MCR REM RSXBAT
$RUN CAMRPT
$FOJ
```

To run the program, mount the CAMS/CAS Interface tape and enter a mount message.

For MTO the message would be:

```
MCR > MOU MTO:/CHA=[FOR]      (CR)
```

Then load the card reader with the CAMRPT Batch deck and enter BAT CR:, to read in the deck.

FORTRAN IV-PLUS V02-04 CALC.FTN /TR:BLOCKS/WR PAGE 1

```
      SUBROUTINE CALC
      IMPLICIT INTEGER(A-Z)
      INCLUDE 'COMMON1.DAT'
*     BYTE CLASFY(4,8),FLDTTL(12)
*     LOGICAL*1 TYPE(20),LABEL(20)
*     LOGICAL*1 FACALT(26),BACLA(26),BUCLAB(26),BULABL(26),GRAINS(26)
*     REAL PCC1,PCC2,PCCG1,PCCG2,PCL1,PCL2,PCLG1,PCLG2
*     REAL CLAD,CLUD,TCONS,RCONS,CCONS
*     REAL BATCON,BUTCON,BARC2,BURCON
*     DIMENSION CLAD(2),CLUD(2),TCONS(2,8),CCONS(2),RCONS(2,8,8)
*     DIMENSION BATCON(2,8),BUTCON(2,8),BARCON(2,8,8),BURCON(2,8,8)
*     DIMENSION ACA(26,26),BUC(26,26)
*     DIMENSION BARPR(30),BURPR(30)
*     COMMON/CBIAS/BARPR,BURPR,BAUNCO,BUUNCO,NZONS,BUCAR,BACAT,BUCAT,
*    *        BAVAR,BUVAR,BARANS,BURANS,A,CC,RH,TC,BUCATS,BUCLAB,
*    *        BULABL,FACALT,CLASFY,TYPE,FLDTTL,BACATS,BACLAB,AUNCO,
*    *        PCC1,PCC2,PCCG1,PCCG2,PCL1,PCL2,PCLG1,PCLG2,APOP,
*    *        CCONS,TCONS,RCONS,CLAD,CLUD,TCONS,RCONS,CCONS,
*    *        BATCON,BUTCON,BARCON,BURCON,FLABEL,GRAINS
*     COMMON/ARRY/CLASFY,FLDTTL
*     COMMON/CLCSM/CLMAT
*     REAL B(3,2)
*     DIMENSION BAUNCO(30),BUUNCO(30),BACAR(30),BUCAR(30)
*     REAL BACAT(26,26),BUCAT(26,26),BAVAR(30),BUVAR(30)
*     REAL BARANS(30),BURANS(30)
      DIMENSION ICAT(12),JCAT(26)
      REAL PSUM,VAR,HI,RLO,BASE1,BASE2,B3F,BASE
      REAL SUNTF,ALPHF,ZETF,VICHF,BAFF,SWSF,ZHF
      REAL PCUR,LOATH(30),SLX,CLX,VARY(8),BARPRF(30)
      BYTE SYMBOL,C(12),FLC,SYMBOL(26)
      LTYP2=0
      NTYP1=0
      NAII=0
      NATI=0
      NGIJ=0
      PASS=22932
      LL=2
      BARPR(26)=APOP
      BAUNCO(26)=AUNCO
      NN=0
      NB=30
      NA=1
      BARPR(NA)=0
      BAVAR(NA)=0
      BARPR(NA)=0
      BAUNCO(NA)=0
            BARANS(NA)=0
      NGIJ=0
      DO 5 I=1,26
    5 BRANCH(I)=0
      M=0
      N=0
      S=0
      PCC1=0
      PCC2=0
      PCL1=0
```

ORIGINAL PAGE IS
OF POOR QUALITY

```
      [illegible]
      [illegible]
      IF(CLADUP.EQ.1) GO TO 470
      DO 15 I=1,NACATS
      IF([illegible]LAR(I).EQ.'[illegible]') [illegible]=1
      IF([illegible]LAR(I).EQ.'S') S=1
      [illegible]
      IF([illegible]LAR(I).[illegible].GRA[illegible]S(J)) GO TO 10
      [illegible]
      GR[illegible](NGLP)=1
      GO TO 15
10    CONTINUE
12    IF([illegible]) GO TO 15
      [illegible]=I
15    CONTINUE
      DO 40 I=1,11
      [illegible](I)=0
40    CONTINUE
      [illegible] J=1,2[illegible]
      DO [illegible] I=1,2[illegible]
      [illegible](J,I)=0
401   CONTINUE
      DO 41 I=1,30
      DATA [illegible](I)=0
41    CONTINUE
      DO 39 I=1,209
      [illegible](I)
      [illegible]ASEY(2*I-1)
      [illegible]ASEY(2*I)
      IF([illegible]) GO TO 3[illegible]
      IF([illegible]) GO TO 39
      IF (TYPE(I).[illegible].'2'.[illegible].TYPE(I).EQ.'0') GO TO 35
      IF ([illegible]) GO TO [illegible]
C
C COMPUTE TYPE 1 [illegible] DATA
C
      [illegible]TYPE1=[illegible]TYPE1+1
      IF ([illegible]) GO TO 32
      [illegible]+1
      GO TO 35
32    IF([illegible].LT.2[illegible]) GO TO 35
      DO 34 II=1,[illegible]
      IF([illegible].NE.GACLA[illegible](II))) GOTO 34
      DO 33 JJ=1,[illegible]
      IF([illegible].EQ.II) GOTO 33
      IF([illegible].NE.GACLA[illegible](JJ))) GOTO 33
      [illegible]+1
33    CONTINUE
34    CONTINUE
C
C COMPUTE TYPE 2 [illegible]
C
35    IF (TYPE(I).NE.'0')GO TO 39
      IF([illegible]) GO TO 3[illegible]
      [illegible]TYPE2=[illegible]TYPE2+1
      IF([illegible]) [illegible]=[illegible]+1
```

```
C
C CONSTRUCT [illegible]IAS CORRECTION VECTORS(LC)
C
      Z1=[illegible]
      Z2=[illegible]
      [illegible] K=1,NACATS
      IF([illegible].EQ.[illegible]([illegible]))Z1=[illegible]
      IF([illegible].EQ.GACLAD([illegible]))Z2=[illegible]
43    CONTINUE
      IF(Z1.EQ.0.[illegible].Z2.EQ.0)GO TO 39
      COVA(Z1,Z2)= COVA(Z1,Z2)+1
39    CONTINUE
C
C COMPUTE [illegible] POPULATION, CLASSIFIED %, CORRECTED %
C
      IF([illegible].LT.2)GO TO [illegible]
C
C COMPUTE [illegible] CORRECTED % AND VARIANCE %
C
      ALC[illegible]=[illegible]
      DO [illegible] II=1,[illegible]
      [illegible](KA)=CAP[illegible](KA)+[illegible](GENLBL(II))
      [illegible](KA)=[illegible](KA)+[illegible](CRNTBL(II))
      DO [illegible] JJ=1,[illegible]
55    [illegible]=[illegible]+[illegible](II),[illegible](JJ))
60    CONTINUE
      ALC[illegible]=[illegible]
      DO 95 I=1,NACATS
      DO 95 II=1,[illegible]
95    ALC[illegible]=ALC[illegible]+COVA(I,GR[illegible](II))
96    CONTINUE
      ALC=[illegible]
      ALCT[illegible]=FLOAT(ALCT)
      ALCF[illegible]=FLOAT(ALCF)
      IF(AL[illegible].GT.[illegible])ALC=ALCTF/ALCF
      ALN[illegible]=[illegible]
      [illegible]=[illegible]
      DO 125 I=1,NACATS
      DO 115 II=1,[illegible]
      IF(I.[illegible].GENL[illegible](II)) GO TO 125
115   CONTINUE
      DO [illegible] J=1,NACATS
      DO [illegible] II=1,[illegible]
      IF(J.EQ.[illegible](II)) GO TO 125
120   CONTINUE
      [illegible]T=[illegible]T+COVA(I,J)
125   CONTINUE
      DO [illegible] I=1,NACATS
      [illegible] J=1,NACATS
      [illegible] II=1,[illegible]
      IF([illegible].EQ.GN[illegible](II)) GO TO 150
148   CONTINUE
      AL[illegible]=[illegible]+COVA(I,J)
150   CONTINUE
      [illegible]=FLOAT([illegible])
      [illegible]=[illegible].
```

```
              BAPF=FLOAT(B[illegible])
              IF(BASEF.GT.0) BASE1=BAPF/BASEF
              ALNTF=FLOAT(ALNT)
              ALNBF=FLOAT(ALNB)
              ALN=0.
              IF(ALNBF.GT.0) ALN=ALNTF/ALNBF
              N3F=0.
              DO 156 I=1,BACATS
              DO 155 II=1,NGM
              IF(I.EQ.GRNLBL(II)) GOTO 159
     155      CONTINUE
              N3F=N3F+FLOAT(BAP2P(I))
     159      CONTINUE
              BACMP(KA)=0.
              BASEP2=0.
              IF(BASEF.LE.0) GO TO 161
              BACMP(KA)=(ALG#BASE1*(I,*ALN)*N3F/BASEF)*100.
              BASEP2=(N3F/BASEF*100.)**2
     161      Z4=0
              DO 21 J=1,BACATS
              DO 20 II=1,NGLM
     20       Z4=Z4+BCVA(J,GRNLBL(II))
     21       CONTINUE
              VAR=0.
              Z4F=FLOAT(Z4-1)
              IF (Z4.LE.1)GO TO 24
              VAR=((BASE1*100.)**2*(ALG*(I,*ALG)))/Z4F
     24       IF(ALNBF.GT.0.) GO TO 22
              BAVAR(KA)=VAR
              GO TO 25
     22       BAVAR(KA)=VAR+BASEP2*(ALN*(I,*ALN))/(ALNBF-1.)
C
C COMPUTE THE RANDOM SAMPLE FOR EACH CATEGORY EXCEPT FOR GRAIN
C
     25       NRATOT=0
              PSUM=0.
              DO 160 I=1,BACATS
              PSUM=PSUM+FLOAT(BAP2P(I))
     160          CONTINUE
              PSUM=PSUM/(22932.-BAP2P(LNI))
              SAMPLE=FLOAT(NSPL)
              DO 165 I=1,BACATS
              BCVT(I)=0
              DO 168 J=1,BACATS
              BCVT(I)=BCVT(I)+BCVA(I,J)
     168      CONTINUE
              BARANS(I)=0.
              IF(SAMPLE.GT.0) BARANS(I)=FLOAT(BCVT(I))/SAMPLE*PSUM*100.
              NRATOT=NRATOT+BCVT(I)
              DO 163 II=1,NGLM
     163      IF(I.EQ.GRNLBL(II)) NGR=NGR+BCVT(I)
     165      CONTINUE
C
C COMPUTE THE RANDOM SAMPLE FOR GRAIN
C
              IF(NGLM.LT.2)GOTO 130
```

ORIGINAL PAGE IS
OF POOR QUALITY

```
0199            GASF=FLOAT(G[illegible]S)
0200            IF(N[illegible]CLF.GT.0) GARANS(KA)=GASF/N[illegible]CLF*PS[illegible]M*100.
          C
          C COMPUTE PCC1, PCC2, PCCG1, PCCG2
0201      100   BATAT=0
0202            X=0
0203            IF ([illegible]TYP1 .EQ. [illegible]) GO TO 45
0204            PCC1=FLOAT(NA[illegible]I)*100./FLOAT(NTYP1)
0205      45    IF([illegible]TYP2 .EQ. [illegible]) GO TO 469
0206            DO 50 I=1,8
0207      50    X=X+[illegible]VA(I,I)
0208            PCC2=FLOAT(X)*100./FLOAT(NTYP2)
0209      469   IF([illegible].LT.2)GO TO 470
0210            DO 174 II=1,[illegible]
0211            DO 173 JJ=1,[illegible]
0212            IF(II.EQ.JJ) GOTO 173
0213            BATAT=BATAT+[illegible]VA(GR[illegible]LBL(II),GRNLBL(JJ))
0214      173   CONTINUE
0215      174   CONTINUE
0216            DO 175 I=1,[illegible]CATS
0217            BATAT=BATAT+[illegible]VA(I,I)
0218      175   CONTINUE
0219                IF ([illegible]TYP1.[illegible])GO TO 468
0220            PCCG1=(FLOAT([illegible]II)+FLOAT(N[illegible]IJ))*100./FLOAT(NTYP1)
0221      468   IF([illegible]TYP2.EQ.0)GO TO 470
0222            PCCG2=FLOAT(BATAT)*100./FLOAT(NTYP2)
0223      470   CONTINUE
          C CLASS CALCULATIONS FIRST
0224            DO [illegible] [illegible]=1,2
0225            IF([illegible].EQ.1) [illegible]K=[illegible]
0226            IF([illegible].EQ.2) [illegible]K=[illegible]
0227            SL[illegible]=[illegible].
0228            SL[illegible]=0.
0229            DO 230 LL=1,2
0230            DO 230 IL=1,8
0231            RAT[illegible](LL,IL)=0.
0232            [illegible](LL,IL)=0.
0233            DO 230 LI=1,8
0234            RAT[illegible](LL,IL,LI)=0.
0235            RUR[illegible](LL,IL,LI)=0.
0236      230   CONTINUE
0237            IF([illegible].EQ.0) GO TO 27[illegible]
0238            VARY(7)=0.
0239            VARY(1)=[illegible]ARC[illegible](K[illegible])
0240            VARY(2)=[illegible](K[illegible])
0241            VARY(3)=[illegible]ATA[illegible]S(K[illegible])
0242            VARY(4)=[illegible]AVAT(KK)
0243            VARY(5)=[illegible]ACAIN(KK,KK)
0244            [illegible]
0245            IF([illegible]) VARY(7)=[illegible]CAI[illegible](N,N)
0246            VARY([illegible])=PCC2
0247            DO 2[illegible] I=1,8
0248            RATC[illegible](K[illegible],I)=T[illegible]S([illegible],I)*VARY(I)
0249            SL[illegible]=SL[illegible]+RATC[illegible]([illegible],I)
0250            DO 2[illegible] J=I,8
0251            PARC[illegible](K[illegible],I,J)=[illegible]S([illegible],I,J)*VARY(I)*VARY(J)
```

```
      CL  =CL  N+TARC  N(   ,  ,J)
 240  CONTINUE
 250  CONTINUE
      SL  =SL  +PCONS(W )
 270  CLAP(W )=CLW+SL
 280  CONTINUE
      TYP =0
      NAII=0
      NPII=0
      NGIJ=0
      BASE=02932
      GWS=0
      BU  R(WU)=0
      BUVAR(WU)=0
      BUCAR(WU)=0
      BUUNC (WU)=0
      BURATS(WU)=0
      NGLM=0
      DO 900 I=1,25
 900  GRN   (I)=0
      X=0
      S=0
      PCL1=0
      NPCL=0
      TYPI=0
      PCL2=0
      PCLC1=0.
      PCLC2=0.
      IF(CL  DUM.EQ.1) RETURN
      DO 915 I=1,B  CATS
      IF(  FLA (I).E .'  ') X=1
      IF(  FLA (I).EQ.'S') S=1
      DO 910 J=1,NLAGS
      IF(  FLA (I).NE.  A  S(J)) GO TO 910
      NGLM=NGLM+1
      GRN   (NGLM)=I
      GOTO 915
 910  CONTINUE
 915  CONTINUE
      DO 940 I=1,11
      LCAT(I)=0
 940  CONTINUE
      DO 9401 J=1,26
      DO 9401 I=1,26
      RCV (J,I)=0
 9401 CONTINUE
      DO 941 I=1,3
      BUR  S(I)=0
 941  CONTINUE
      DO 939 I=1,2
      LR=  FD(I)
      CL=CL  T(2*I-1)
      CL =CL DAT(2*I)
      IF(L  .EQ.  ) GO TO 939
      IF (   .EQ.   .   .EQ.  ) GO TO 939
      IF (TYPE(I).EQ.'P'. R.TYPE(I).EQ.'R') GO TO 935
```

```
      IF (CL2.EQ.' ') GO TO 939
C
C COMPUTE TYPE 1 DOT DATA
C
      NTYP1=NTYP1+1
      IF (CLP.NE.CL .AND. CL.NE.' ') GO TO 932
      NATT=NATT+1
      GO TO 935
932   IF(NSUM.LT.2)GO TO 935
      DO 934 II=1,NSUM
      IF(CLP.NE.BRCLAS(GRNLBL(II))) GOTO 934
      DO 933 JJ=1,NSUM
      IF(JJ.EQ.II) GOTO 933
      IF(CL.NE.BRCLAS(GRNLBL(JJ))) GOTO 933
      NSIJ=NSIJ+1
933   CONTINUE
934   CONTINUE
C
C COMPUTE TYPE 2 DOT
C
935   IF (TYPE(1).NE.'2')GO TO 939
      IF(CLP.EQ.' ') GO TO 939
      NTYP2=NTYP2+1
      IF(CL.NE.' '.AND.CL.NE.' ') N2CL=N2CL+1
C
C CONSTRUCT BIAS CORRECTION VECTORS(LE)
C
      Z1=0
      Z2=0
      DO 943 K=1,NCATS
      IF(CLP.EQ.BRCLAS(K))Z1=K
      IF(CL.EQ.BRCLAS(K))Z2=K
943   CONTINUE
      IF(Z1.EQ.0.OR.Z2.EQ.0)GO TO 939
      BCV(Z1,Z2)=BCV(Z1,Z2)+1
939   CONTINUE
C
C COMPUTE GRAIN POPULATION, CLASSIFIED %, CORRECTED %
C
      IF(NSUM.LT.2)GO TO 925
C
C COMPUTE GRAIN CORRECTED % AND VARIANCE %
C
      ALCT=0
      DO 960 II=1,NSUM
      PUP(WD)=PUP(WD)+[illegible](GRNLBL(II))
      [illegible](WD)=[illegible](WD)+[illegible](GRNLBL(II))
      DO 955 JJ=1,NSUM
955   ALCT=ALCT+COV([illegible](II),GRNLBL(JJ))
960   CONTINUE
      ALCT=0
      DO 970 I=1,NCATS
      DO 975 II=1,NSUM
975   ALCT=ALCT+[illegible](I,GRNLBL(II))
970   CONTINUE
      ALCT=FLOAT(ALCT)
```

ORIGINAL PAGE IS
OF POOR QUALITY

```
      ALGBF=FLOAT(ALGB)
      ALG=0.
      IF(ALGBF.GT.0) ALG=ALGTF/ALGBF
      ALNB=0
      ALNT=0
      DO 9125 I=1,NUCATS
      DO 9115 II=1,NGL
      IF(I.EQ.GRPLGL(II)) GOTO 9125
9115  CONTINUE
      DO 9125 J=1,NUCATS
      DO 9120 II=1,NGL
      IF(J.EQ.GRPLGL(II)) GOTO 9125
9120  CONTINUE
      ALNT=ALNT+COVU(I,J)
9125  CONTINUE
      DO 9150 I=1,NUCATS
      DO 9150 J=1,NUCATS
      DO 9145 II=1,NGLN
      IF(J.EQ.GRPLGL(II)) GOTO 9150
9145  CONTINUE
      ALNB=ALNB+COVU(I,J)
9150  CONTINUE
      BASEF=FLOAT(BASE-SUPPR(GU?))
      BASE1=0.
      SAPF=FLOAT(SUPPR(?))
      IF(BASEF.GT.0) BASE1=SAPF/BASEF
      ALNTF=FLOAT(ALNT)
      ALNBF=FLOAT(ALNB)
      ALN=0.
      IF(ALNBF.GT.0) ALN=ALNTF/ALNBF
      NSF=0.
      DO 9159 I=1,NUCATS
      DO 9155 II=1,NGLN
      IF(I.EQ.GRPLGL(II)) GOTO 9159
9155  CONTINUE
      NSF=NSF+FLOAT(SUPPR(I))
9159  CONTINUE
      PCTR(GU)=0.
      BASE2=0.
      IF(BASEF.LE.0) GO TO 9161
      PCTR(GU)=(ALG*(BASE1+(1.-ALN)*NSF/BASEF)*100.
      BASE2=(NSF/BASEF*100.)**2
9161  Z4=0
      DO 921 J=1,NUCATS
      DO 920 II=1,NGLN
920   Z4=Z4+COVU(J,GRPLGL(II))
921   CONTINUE
      VAR=0.
      Z4F=FLOAT(Z4-1)
      IF (Z4.LE.1) GO TO 924
      VAR=((BASE1*100.)**2*(ALG*(1.-ALG)))/Z4F
924   IF(ALNBF .GT. 1.) GO TO 922
      PCTVAR(GU)=VAR
      GO TO 925
922   PCTVAR(GU)=VAR+BASE2*ALN*(1.-ALN)/(ALNBF-1.)
      C
```

```
              C COMPUTE THE RANDOM SAMPLE FOR EACH CATEGORY EXCEPT FOR GRAIN
              C
0404     925            BUTTI=0
0405              PSUM=0.
0406              DO 9160 I=1,NUCATS
0407              PSUM=PSUM+FLOAT(NUPOP(I))
0408     9160     CONTINUE
0409              PSUM=PSUM/(22930.-NUPOP(UNI))
0410              [illegible]=FLOAT([illegible])
0411              DO 9165 I=1,NUCATS
0412              FCVT(I)=0
0413              DO 9168 J=1,NUCATS
0414              FCVT(I)=FCVT(I)+[illegible](I,J)
0415     9168     CONTINUE
0416              NURANS(I)=0.
0417              IF([illegible]CLF.GT.0) NURANS(I)=FLOAT(FCVT(I))/NOCLF*PSUM*100.
0418              [illegible]=[illegible]+FCVT(I)
0419              DO 9163 II=1,[illegible]
0420     9163     IF(I.EQ.[illegible](II)) [illegible]AS+FCVT(I)
0421     9165     CONTINUE
              C
              C COMPUTE THE RANDOM SAMPLE FOR GRAIN
              C
0422              IF([illegible].LT.2)GOTO 9100
0423              [illegible]=FLOAT([illegible])
0424              NURANS([illegible])=0.
0425              IF([illegible]CLF.GT.0)NURANS([illegible])=GASF/NOCLF*PSUM*100.
              C
              C COMPUTE [illegible], PC02, [illegible], [illegible]
0426     9100     [illegible]=0
0427              X=[illegible]
0428              IF ( [illegible]TYP1 .EQ. 0) GO TO 945
0429              PC[illegible]=FLOAT([illegible])*100./FLOAT(NTYP1)
0430     945      IF( [illegible]TYP2 .EQ. 0) GO TO 9469
0431              DO [illegible] I=1,5
0432     950      X=X+[illegible]VU(I,I)
0433              [illegible]=FLOAT(X)*[illegible]/FLOAT(NTYP2)
0434     9469      IF( [illegible].LT.2)[illegible] TO 9470
0435              DO [illegible] II=1,[illegible]
0436              DO 9173 JJ=1,[illegible]
0437              IF(II.EQ.JJ)GOTO 9173
0438              [illegible]=[illegible]ATAT+[illegible](GR[illegible](II),GRNLRL(JJ))
0439     9173     CONTINUE
0440     9174     CONTINUE
0441              DO 9175 I=1,[illegible]
0442              [illegible]=[illegible]ATAT+[illegible](I,I)
0443     9175     CONTINUE
0444                IF ([illegible]TYP1.EQ.0)GO TO 9468
0445              [illegible]=(FLOAT([illegible])[illegible])*100./FLOAT(NTYP1)
0446     9468      IF( [illegible]TYP2.EQ.0)GO TO 9470
0447              [illegible]=FLOAT([illegible])*[illegible]/FLOAT(NTYP2)
0448     9470     CONTINUE
              [illegible]
0449              DO [illegible] [illegible]=1,2
0450              IF[illegible]
0451              IF([illegible].EQ.2) [illegible]
```

```
      SL[illegible]=0.
      CL[illegible]=0.
      IF([illegible]) GO TO 9270
      WARY(1)=1.
      WARY(2)=[illegible](KK)
      WARY(3)=[illegible](KK)
      WARY(4)=[illegible](KK)
      WARY(5)=[illegible](KK)
      WARY(6)=[illegible](KK,[illegible])
      WARY(7)=PCI1
      IF([illegible]) WARY(7)=[illegible](7,N)
      WARY(8)=PCI2
      DO 9250 I=1,8
      PLTC[illegible](KK,I)=T[illegible]S([illegible],I)*WARY(I)
      SL[illegible]=SL[illegible]+PLTC[illegible](KK,I)
      DO 9240 J=1,8
      PURC[illegible](KK,I,J)=[illegible](KK,I,J)*WARY(I)*WARY(J)
      CL[illegible]=CL[illegible]+PURC[illegible](KK,I,J)
9240  CONTINUE
9250  CONTINUE
      SL[illegible]=SL[illegible]+[illegible](KK)
9270  CL[illegible]([illegible])=CL[illegible]+SL[illegible]
9280  CONTINUE
      CONTINUE
      RETURN
      END
```

FORTRAN IV [illegible] [illegible] [illegible]

CALC.[illegible] [illegible]/[illegible]

PAGE 11

PROGRAM SECTIONS

NUMBER	NAME	SIZE		ATTRIBUTES
1	[illegible]	[illegible]	2675	[illegible],LCL
2	[illegible]	[illegible]	2	[illegible],LCL
4	[illegible]	[illegible]	[illegible]	[illegible],LCL
5	[illegible]	[illegible]	22	[illegible],LCL
6	CE[illegible]	[illegible]	[illegible]	[illegible],VR,GBL
7	[illegible]	[illegible]	2	[illegible],VR,GBL
8	[illegible]	[illegible]	2[illegible]5	[illegible],VR,GBL

ENTRY POINTS

NAME	TYPE	ADDRESS	NAME	TYPE	ADDRESS	NAME	TYPE	ADDRESS	NAME	TYPE	ADDRESS	NAME	TYPE	ADDRESS
CALC		1-[illegible]												

VARIABLES

NAME	TYPE	ADDRESS	NAME	TYPE	ADDRESS	NAME	TYPE	ADDRESS	NAME	TYPE	ADDRESS	NAME	TYPE	ADDRESS
AL[illegible]	[illegible]	[illegible]	AL[illegible]	I*2	[illegible]	AL[illegible]	[illegible]	[illegible]	AL[illegible]	I*2	4-0[illegible]0770	ALCYF	R*4	4-000[illegible]2
AL	R*4	4-00[illegible]72	AL[illegible]	I*2	4-00[illegible]774	AL[illegible]F	R*4	4-000156	ALAY	I*2	4-000776	ALAYF	R*	-000152
A[illegible]	I*[illegible]	[illegible]	A[illegible]	I*2	[illegible]	DACATS	I*2	6-015[illegible]32	BAFF	R*4	4-000172	BASF	I*2	4-000726
BAS[illegible]	R*4	4-00[illegible]	BAS[illegible]	R*4	4-000132	BASF2	R*4	4-000136	BATAT	I*2	4-001004	BATCT	I*6	6-023352
[illegible]ATS	I*[illegible]	[illegible]	[illegible]	I*2	[illegible]356	[illegible]	[illegible]	[illegible]	[illegible]	I*2	7-[illegible]	CUOMUM	I*2	7-000002
CL[illegible]	R*4	4-00[illegible]	CL2	L*1	4-00[illegible]6[illegible]	D[illegible]	I*2	6-014200	DJ	I*2	6-014202	GWS	I*2	4-000732
GA[illegible]	[illegible]	[illegible]	I	I*2	[illegible]741	II	I*2	4-000756	IT	I*2	4-001016	J	I*2	4-000754
JJ	I*2	4-00[illegible]	K	I*2	4-000766	KK	I*2	4-001012	LP	L*1	4-000642	LT	I*2	4-001020
LL	I*2	[illegible]	[illegible]	I*2	4-001010	N	I*2	4-000744	NAIT	I*2	4-000720	NBIT	I*[illegible]	4-000772
NCIJ	I*2	4-00[illegible]	NCL[illegible]	I*2	4-00[illegible]74	NLANG	I*2	6-026706	NOCL	I*2	4-000762	NOCLF	R*4	4-000206
NTYPE	I*2	4-00[illegible]	NTYP2	I*2	4-000714	NSF	R*4	4-000142	P	I*2	4-001002	PCCA1	R*4	6-016100
PCOS2	R*4	6-01[illegible]	PCC[illegible]	R*4	6-01[illegible]7[illegible]	PCC2	R*4	6-016074	PCLG1	R*4	6-016120	PCLG2	R*4	6-016124
PCLT	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	S	I*[illegible]	4-000750	SLW	R*4	4-00[illegible]
TC	I*2	4-00[illegible]	UN1	I*2	4-00[illegible]73[illegible]	VAR	R*4	4-000116	W	I*2	4-000746	WA	I*2	4-000736
WIT	I*2	4-00[illegible]	X	I*2	4-00[illegible]	Z1	I*2	4-000752	Z2	I*2	4-000764	Z4	I*2	4-001010
ZAF	R*4	4-00[illegible]2												

ARRAYS

NAME	TYPE	ADDRESS	SIZE		DIMENSIONS
A	R*4	[illegible]	[illegible]	[illegible]	(2,2)
B[illegible]	R*4	[illegible]	[illegible]	[illegible]	(2,[illegible])
[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	([illegible])
[illegible]	I*[illegible]	[illegible]	[illegible]24	3[illegible]	([illegible])
[illegible]	I*[illegible]	[illegible]	[illegible]	[illegible]	([illegible])
BA[illegible]	I*2	[illegible]	[illegible]	[illegible]	([illegible])
B[illegible]	R*4	[illegible]	[illegible]	[illegible]	([illegible])
BA[illegible]	R*4	[illegible]	[illegible]	[illegible]	(3[illegible])
[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	([illegible])
BATC[illegible]	R*4	[illegible]	[illegible]	[illegible]	([illegible])

[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]
BAKER	[illegible]	[illegible]	[illegible]	[illegible]	(2)
BC[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	(2,26)
BCVT	[illegible]	[illegible]	[illegible]	[illegible]	(2)
BCVC	1*3	6=027410	000513	676	(26,26)
BLCAIN	[illegible]	[illegible]	[illegible]	[illegible]	(2,26)
[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]
BL[illegible]R	[illegible]	[illegible]	[illegible]	[illegible]	(3)
BLLABL	[illegible]	[illegible]	[illegible]	33	(20)
BLRCH	[illegible]	[illegible]	[illegible]	[illegible]	(4)
BURLNS	[illegible]	[illegible]	[illegible]	[illegible]	(30)
BURTI	[illegible]	[illegible]	[illegible]	[illegible]	(2,8,8)
[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]
BULLCT	[illegible]	[illegible]	[illegible]	[illegible]	(30)
BUJJR	[illegible]	[illegible]	[illegible]	[illegible]	(30)
C	[illegible]	[illegible]	[illegible]	[illegible]	(12)
CERNS	[illegible]	[illegible]	[illegible]	[illegible]	(2)
CLAM	[illegible]	[illegible]	[illegible]	4	(2)
[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]
CLRRT	[illegible]	[illegible]	[illegible]	[illegible]	(41)
CL[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	(2)
CPAINS	[illegible]	[illegible]	[illegible]	13	(26)
CP[illegible]	[illegible]	[illegible]	[illegible]	13	[illegible]
L[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	(2,9)
[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]
LCATC	[illegible]	[illegible]	[illegible]	5	(2)
RCPTS	[illegible]	[illegible]	[illegible]	[illegible]	(2,8,8)
TCNS	[illegible]	6=02[illegible]74	[illegible]	32	(2,6)
TYPE	[illegible]	[illegible]	[illegible]	[illegible]	(2,9)
WARY	[illegible]	4=00[illegible]	[illegible]	1[illegible]	()

LABELS

LABEL	ADDRESS	LABEL	ADDRESS	LABEL	ADDRESS	LABEL	ADDRESS	LABEL	ADDRESS
5	**	1[illegible]	[illegible]	12	**	15	1=00[illegible]604	20	**
21	**	22	1=[illegible]	24	1=003322	25	1=003454	32	1=001104
33	1=[illegible]124[illegible]	34	1=[illegible]	35	1=00[illegible]	39	1=001514	40	**
41	**	4[illegible]	**	45	1=004176	50	**	55	**
6[illegible]	**	9[illegible]	**	95	**	100	1=004124	115	**
12[illegible]	**	125	1=[illegible]	140	**	150	1=012454	155	**
159	1=[illegible]2754	16[illegible]	**	171	1=[illegible]14	183	**	185	**
16[illegible]	**	173	1=[illegible]	174	**	175	**	239	**
24[illegible]	**	2[illegible]	**	27[illegible]	1=003[illegible]	280	**	4[illegible]1	**
469	1=[illegible]4630	46[illegible]	1=[illegible]	47[illegible]	1=[illegible]444A	9[illegible]0	**	910	1=006176
94[illegible]	1=[illegible]2[illegible]	[illegible]	**	921	**	9[illegible]2	1=011406	924	1=011[illegible]4[illegible]
92[illegible]	1=[illegible]117[illegible]	9[illegible]2	[illegible]	9[illegible]3	1=006754	9[illegible]4	1=006776	935	1=007020
939	1=[illegible]25[illegible]	[illegible]	**	941	**	943	**	945	1=011[illegible]12
95[illegible]	**	[illegible]5	**	9[illegible]0	**	995	**	99[illegible]	**
91[illegible]	1=[illegible]	91[illegible]	**	912[illegible]	**	91[illegible]	1=01[illegible]005	9145	**
915[illegible]	1=[illegible]0171	91[illegible]	**	915[illegible]	1=01[illegible]	916[illegible]	**	9161	1=0[illegible]0634
91[illegible]	**	916[illegible]	**	9160	**	9173	1=012155	9174	**
917[illegible]	**	9[illegible]	**	9[illegible]	**	927[illegible]	1=0[illegible]	9280	**
98[illegible]	**	9[illegible]	1=[illegible]	9[illegible]	1=01[illegible]028	947[illegible]	1=012[illegible]		

TOTAL SPACE ALLOCATED = 041100 9248

.LPI=CALC

```
      SUBROUTINE BIAPRT(ITYPE,TAPENO,PASS)
      IMPLICIT INTEGER (A-Z)
      LOGICAL*1 IBUF(1)
      [illegible]CAT'
      BYTE CLASFY(418),[illegible](418)
      LOGICAL*1 TYPE(209),LBLED(209)
      [illegible](26),[illegible](26),BUCLAB(26),BULABL(26),GRAINS(26)
      REAL [illegible]1,[illegible],PCLG2,PCL1,PCL2,PCLG1,PCLG2
      REAL [illegible],[illegible]S,CCZNS
      REAL [illegible],[illegible],RCZN,BURCZN
      DIMENSION [illegible](2),[illegible](2),TCZNS(2,8),CCZNS(2),RCZNS(2,8,8)
      DIMENSION [illegible](2,8),RARCZN(2,8,8),BURCZN(2,8,8)
      DIMENSION [illegible],PCVL(26,26)
      DIMENSION [illegible], UPAR(30)
      COMMON /CBIAS/[illegible],PTP,BACNCZ,BUUNCZ,BACZR,BUCZR,BACAIN,BUCAIN,
     *          [illegible]AR,RARANS,BURANS,A,[illegible],TC,BUCATS,BUCLAB,
     *          [illegible]LABL,CLASFY,TYPE,LBLED,RACATS,BACLAB,AUNCR,
     *          [illegible],PCCG2,PUCG2,PCL1,PCL2,PCLG1,PCLG2,APZP,
     *          [illegible],CLAD,CLUD,TCZNS,RCZNS,CCZNS,
     *          [illegible],RARCZN,BURCZN,LABS,GRAINS
      COMMON [illegible]
      COMMON /BLCOM/[illegible]
      REAL A(3,2)
      DIMENSION [illegible](30),BUUNCZ(30),RACOR(30),BUCZR(30)
      REAL [illegible](26,26),BAVAR(30),BUVAR(30)
      REAL RARANS(30), [illegible]S(30)
      COMMON /SEG/[illegible]
      LOGICAL*1 [illegible](120),BLANK,PMATR(120)
      DIMENSION [illegible](19)
      DATA [illegible]/10,20,30,40,50,60,70,80,90,100,110,120,130,140,
     *         150,160,170,180,190/
      DATA BLANK/' '/,[illegible]/'1'/,SLASH/'/'/,TWO/'2'/
  100 FORMAT(1H1,44X,'[illegible] CORRECTION CLASSIFICATION REPORT')
  101 FORMAT(1H0,2[illegible](5X,'CAT'))
  102 FORMAT(1H ,17X,1[illegible]X,I3))
  103 FORMAT(1H0,5X,'[illegible]      ',10(3X,I5))
  104 FORMAT(1H ,5X,'[illegible]   % ',10(2X,I6))
  105 FORMAT(1H ,5X,'[illegible]   % ',10(2X,I6))
  106 FORMAT(1H ,5X,'[illegible]   % ',10(2X,F6.2))
  107 FORMAT(1H ,5X,'[illegible] SAMPLE ',10(2X,F6.2))
  108 FORMAT(1H1,51X,'[illegible] VALUE MATRIX')
  109 FORMAT(1H ,14X,'<[illegible]> CAT',I3))
  110 FORMAT(1H ,1X,'[illegible]',7'I3(6X,1A1,1X))
  111 FORMAT(1H ,1X,'[illegible]')
  112 FORMAT(1H ,11X,[illegible],13(4X,F6.3))
  113 FORMAT(1H1,3[illegible] CORRECTION VECTORS (LABLED:CLASSIFIED)')
  114 FORMAT(1H0,14X,8(4X,1A1,':',1A1,I3))
  115 FORMAT('1',42X,'[illegible] INTERFACE REPORT')
  116 FORMAT(1H0,3X,'[illegible] = ',24A1,57X,'TAPE NUMBER: ',I5,'0')
  117 FORMAT(1H0,3X,'[illegible] ID = ',6A1,1X,6A1)
  118 FORMAT(1H0,57X,'[illegible]SITIZN DATES')
  119 FORMAT(1H ,47X,'1',6X,'2',6X,'3',6X,'4')
  120 FORMAT(1H ,4X,'*****  SEGMENT NUMBER =',4A1,13X,4(5A1,2X))
  121 FORMAT(1H ,'[illegible] SEGMENT NO. ',I1,' =',66A1)
  122 FORMAT(1H0,[illegible],'TYPE',1A1,' [illegible] LABEL/CLASSIFICATION')
  123 FORMAT(1H0,[illegible],'TYPE',1A1,' [illegible] LABEL/CLUSTER ASSIGNMENT')
```

```
124     FØRMAT(1H0,1X,'FIXED'/)
125     FØRMAT(' ',1X,'LINE',2X,19(I3,3X))
126     FØRMAT(1H+,74X,'TC',6X,'TH',6X,'UN',3X,'CØMBØ')
127     FØRMAT(1H+,74X,'TC',5X,'CA',6X,'UN',3X,'CØMB7')
128     FØRMAT(1H1,41X,'BIAS CØRRECTIØN CLUSTER REPØRT')
129     FØRMAT(1H1,12X)
130     FØRMAT(1H0,3X,'PCC1   = ',F6.1,3X,' PCC2   = ',F6.1,6X
      1 'PCCS1 = ',F5.1,7X,' PCCS2 = ',F6.1,6X,
      2' DW    = ',F6.1,6X' IS    = ',F6.1)
132     FØRMAT(1H0,12X)
133     FØRMAT(1H+,70X,'CLUSTERED ')
134     FØRMAT(' ',15X,13(A1,'||',A1,I3,2X))
135     FØRMAT(1H0,22X,'TØTAL = ',I6)
145     FØRMAT(1H ,5X,'CØRRECTED      % ',6(2X,I6),26X,I6)
146     FØRMAT(1H ,5X,'VARIANCE       % ',6(2X,F6.2),26X,F6.2)
147     FØRMAT(1H ,5X,'RANDØM SAMPLE   ',6(2X,F6.2),26X,F6.2)
148     FØRMAT(1H+,102X,'VARIANCE IS UNDEFINED')
150     FØRMAT(1H0,45X,'INTERMEDIATE VALUES FØR DW AND DS')
151     FØRMAT(1H ,36X,' W',58X,'DS'/6X,'P=',4X,8I6,6X,'P=',4X,8I6)
152     FØRMAT(1H ,2X,'T(P)W(P)  ',8F6.3,'  V(P)S(P)  ',8F6.3)
153     FØRMAT(1H ,29X,'P(P,Q)W(P)W(Q)',46X,'U(P,Q)S(P)S(Q)')
154     FØRMAT(1H ,2X,'Q=',I4,4X,8F6.2,12X,8F6.2)
155     FØRMAT(1H ,15X,' ØFFSET CØNSTANT FØR DW =',F10.5,
      1 25X,' ØFFSET CØNSTANT FØR DS =',F10.5)
        GØ TØ (1,2),PASS
1       WRITE(6,115)
        DØ 3 I=1,4
3       SEGNØ(I)=IBUF(147+I)
        WRITE(6,116) (IBUF(I),I=3,26),TAPNØ
        WRITE(6,117) (IBUF(I),I=709,714),(IBUF(J),J=715,720)
        WRITE(6,118)
        WRITE(6,119)
        WRITE(6,120) (SEGNØ(I),I=1,4),(IBUF(J),J=154,158),
      2(IBUF(K),K=159,163),(IBUF(L),L=164,168),(IBUF(M),M=169,173)
        RETURN
2       IF(CLASUM.EQ.1) GØ TØ 202
        WRITE(6,129)
        CNT=..E
        DØ 97 JJ=1,2
        K=1
        NLIM=10
        WRITE(6,122) CNT
        WRITE(6,124)
        WRITE(6,125) (CØL(I),I=1,19)
        DØ 99 J=1,11
        L=6
        DØ 10 I=1,120
        PMATR(I)=BLANK
10      CØNTINUE
        DØ 12 =1,19
        IF(C..T.EQ.'1') GØ TØ 6
        IF(TYPE(K).EQ.'1'.ØR.TYPE(K).EQ.'2') GØ TØ 7
        GØ TØ 5
6       IF(TYPE(K).EQ.'1'.ØR.TYPE(K).EQ.'3') GØ TØ 7
        GØ TØ 5
7       PMATR(L)=LPLED(K)
```

```
      PMATR(L+1)=CLASS
      PMATR(L+2)=CLASSFY(2*K)
      IF(CLASSFY(2*K-1).EQ.[illegible]) GO TO 5
      PMATR(L+2)=CLASSFY(2*K-1)
      PMATR(L+3)=CLASSFY(2*K)
    5 CONTINUE
      K=K+1
      L=L+5
   12 CONTINUE
      WRITE(5,99) [illegible],(PMATR(I),I=6,125)
   99 FORMAT(1H0,2X,I3,2X,115A1)
      NLINE=NLINE+10
   98 CONTINUE
      [illegible]
   97 CONTINUE
      WRITE(6,132)
      WRITE(6,101)
      WRITE(6,102) (RA[illegible]AFL(I),I=1,10)
      WRITE(6,103) (RA[illegible](I),I=1,10)
      WRITE(6,104) (RA[illegible](I),I=1,10)
      WRITE(6,105) (RA[illegible](I),I=1,10)
      WRITE(6,106) (RAVAR(I),I=1,10)
      WRITE(6,107) (RA[illegible]S(I),I=1,10)
      WRITE(6,132)
      WRITE(6,101)
      WRITE(6,102) (RA[illegible]AFL(I),I=11,20)
      WRITE(6,103) (RAPZP(I),I=11,20)
      WRITE(6,104) (RA[illegible](I),I=11,20)
      WRITE(6,105) (RA[illegible](I),I=11,20)
      WRITE(6,106) (RA[illegible](I),I=11,20)
      WRITE(6,107) (RARANS(I),I=11,20)
      WRITE(6,132)
      WRITE(6,101)
      WRITE(6,102) (RA[illegible]AFL(I),I=21,26)
      WRITE(6,126)
      WRITE(6,103) (RAPZP(I),I=21,30)
      WRITE(6,104) (RA[illegible](I),I=21,30)
      WRITE(6,145) ([illegible](I),I=21,26),BACZR(30)
      WRITE(6,146) (RAVAR(I),I=21,26),RAVAR(30)
      IF([illegible]AR(30).EQ.[illegible].AND.RAPZP(30).GT.0)WRITE(6,148)
      WRITE(6,147) (RARANS(I),I=21,26),BARANS(30)
      WRITE(6,132)
      WRITE(6,130) PCC1,PCC2,PCCG1,PCCG2,CLAD(1),CLAD(2)
      WRITE(6,132)
      WRITE(6,150)
      WRITE(6,155) [illegible],COCNS(2)
      WRITE(6,151) (I,I=1,[illegible]),(J,J=1,8)
      WRITE(6,152) (([illegible](I,J),J=1,8),I=1,2)
      WRITE(6,153)
      DO 180 K=1,8
      WRITE(6,154) K,((HARCZN(L,K,M),M=1,8),L=1,2)
  180 CONTINUE
      WRITE(6,132)
      NC=[illegible]
      IF([illegible]ACATS.LE.13) NC=NACATS
      [illegible]
```

```
              WRITE(6,109)
              WRITE(6,109) (I,I=1,NC)
              WRITE(6,110) (CATLAB(I),I=1,NC)
              WRITE(6,111)
              DO 96 J=1,BACATS
              WRITE(6,112) BALABL(J),(BACAIN(J,K),K=1,NC)
      96      CONTINUE
              IF(BACATS.LE.13) GO TO 196
              NC=BACATS
              WRITE(6,132)
              NT=NC-13
              WRITE(6,109) (I,I=14,NC)
              WRITE(6,110) (CATLAB(I),I=14,NC)
              WRITE(6,111)
              DO 196 J=1,BACATS
              WRITE(6,112) BALABL(J),(BACAIN(J,K),K=14,BACATS)
     196      CONTINUE
              NC=13
              IF(BACATS.LE.13) NC=BACATS
              WRITE(6,133)
              DO 50 K=1,BACATS
              WRITE(6,134) (BACLAB(K),BACLAB(LL),BCVA(K,LL),LL=1,NC)
      50      CONTINUE
              WRITE(6,132)
              IF(BACATS.LE.13) GO TO 161
              NC=BACATS
              DO 160 K=1,BACATS
              WRITE(6,134) (BACLAB(K),BACLAB(LL),BCVA(K,LL),LL=14,NC)
     160      CONTINUE
     161      CONTINUE
              WRITE(6,135) DATKT
     202      IF(CLUNUM.EQ.1) RETURN
              WRITE(6,129)
              CNT=0
              DO 207 JJ=1,2
              K=1
              NLINE=0
              WRITE(6,123) CNT
              WRITE(6,124)
              WRITE(6,125) (CTL(I),I=1,19)
              DO 207 J=1,11
              L=6
              DO 210 I=1,120
              PMATR(I)=BLANK
     210      CONTINUE
              DO 207 K=1,19
              IF(CNT.EQ.'1') GO TO 206
              IF(TYPE(K).EQ.'1'.OR.TYPE(K).EQ.'2') GO TO 207
              GO TO 205
     206      IF(TYPE(K).EQ.'1'.OR.TYPE(K).EQ.'3') GO TO 207
              GO TO 205
     207      PMATR(L)=LTLFD(K)
              PMATR(L+1)=SLASH
              PMATR(L+2)=CLDOT(2*K)
              IF(CLDOT(2*K-1).EQ.' ') GO TO 5
              PMATR(L+3)=CLDOT(2*K-1)
```

```
      PMATR(L+3)=CLDST(2*K)
  205 CONTINUE
      K=K+1
      L=L+6
  212 CONTINUE
      WRITE(6,99) NLINE,(PMATR(I),I=6,120)
      NLINE=NLINE+10
  298 CONTINUE
      ONT=TWO
  297 CONTINUE
      WRITE(6,128)
      WRITE(6,101)
      WRITE(6,102) (  ABL(I),I=1,10)
      WRITE(6,103) (  2P(I),I=1,10)
      WRITE(6,104) (  NCR(I),I=1,10)
      WRITE(6,105) (  CR(I),I=1,10)
      WRITE(6,106) (  AR(I),I=1,10)
      WRITE(6,107) (  A S(I),I=1,10)
      WRITE(6,101)
      WRITE(6,102) (  AP (I),I=11,20)
      WRITE(6,103) (  R(I),I=11,20)
      WRITE(6,104) (     (I),I=11,20)
      WRITE(6,105) (  R(I),I=11,20)
      WRITE(6,106) (  AR(I),I=11,20)
      WRITE(6,107) (  A S(I),I=11,20)
      WRITE(6,101)
      WRITE(6,102) (  ABL(I),I=21,26)
      WRITE(6,127)
      WRITE(6,103) (  P(I),I=21,30)
      WRITE(6,104) (  NC (I),I=21,30)
      WRITE(6,145) (  (I),I=21,26),BUC2R(30)
      WRITE(6,146) (  AR(I),I=21,26),BUVAR(30)
      IF(  AR(30).E   0.AND.BLP2P(30).GT.0)WRITE(6,148)
      WRITE(6,147) (  ANS(I),I=21,26),BURANS(30)
      WRITE(6,132)
      WRITE(6,13 ) P  ,PCL2,PCLS1,PCLS2,CLUD(1),CLUD(2)
      WRITE(6,132)
      WRITE(6,15 )
      WRITE(6,155) CC  S(1),CC2NS(2)
      WRITE(6,151) (I,I=1, ),(J,J=1,8)
      WRITE(6,152) ((  TC (I,J),J=1,8),I=1,2)
      WRITE(6,153)
      DO 190 K=1,6
      WRITE(6,154) K,((BU  2N(L,K,M),M=1,8),L=1,2)
  190 CONTINUE
      WRITE(6,132)
      WRITE(6,132)
      NC=13
      IF(  CATS.LE.13) NC=  JCATS
      NT=
      WRITE(6,108)
      WRITE(6,109) (I,I=1,NC)
      WRITE(6,110) (  LA (I),I=1,NC)
      WRITE(6,111)
      DO    J=1,  CATS
      WRITE(6,112)   APL(J),(BUCAT (J,K),K=1,NC)
```

ORIGINAL PAGE IS
OF POOR QUALITY

```
  20    CONTINUE
        IF(BUCATS.LE.13) GO TO 421
        NC=BUCATS
        WRITE(6,112)
        NT=NC-13
        WRITE(6,109) (I,I=14,NC)
        WRITE(6,110) (BUCLAB(I),I=14,NC)
        WRITE(6,111)
        DO 420 J=1,BUCATS
        WRITE(6,112) BUCLAB(J),(BUCAIN(J,K),K=14,BUCATS)
  420   CONTINUE
  421   CONTINUE
        NC=13
        IF(BUCATS.LE.13) NC=BUCATS
        WRITE(6,113)
        DO 250 K=1,BUCATS
        WRITE(6,134) (BUCLAB(K),BUCLAB(LL),BCVU(K,LL),LL=1,NC)
  250   CONTINUE
        WRITE(6,132)
        IF(BUCATS.LE.13) GO TO 450
        NC=BUCATS
        DO 450 K=1,BUCATS
        WRITE(6,134) (BUCLAB(K),BUCLAB(LL),BCVU(K,LL),LL=14,NC)
  450   CONTINUE
        WRITE(6,135) BUT T
        RETURN
        END
```

ORIGINAL PAGE IS
OF POOR QUALITY

PRØGRAM SECTIØNS

NUMBER	NAME	SIZE		ATTRIBUTES
1	$CØDE1	017710	3044	RW,I,CØN,LCL
2	$PDATA	[illegible]	3	RW,D,CØN,LCL
3	$IDATA	[illegible]	674	RW,D,CØN,LCL
4	$VARS	[illegible]	155	RW,D,CØN,LCL
5	$TEMPS	[illegible]	3	RW,D,CØN,LCL
6	CBIAS	[illegible]	5873	RW,D,ØVR,GBL
7	DUMMY	[illegible]	2	RW,D,ØVR,GBL
8	CLCØM	[illegible]	259	RW,D,ØVR,GBL
9	SEG	[illegible]	2	RW,D,ØVR,GBL

ENTRY PØINTS

NAME	TYPE	ADDRESS	NAME	TYPE	ADDRESS	NAME	TYPE	ADDRESS	NAME	TYPE	ADDRESS	NAME	TYPE	ADDRESS
BIAPRT		1-000000												

VARIABLES

NAME	TYPE	ADDRESS	NAME	TYPE	ADDRESS	NAME	TYPE	ADDRESS	NAME	TYPE	ADDRESS	NAME	TYPE	ADDRESS
AP2P	I*2	6-01[illegible]	AJNCA	I*2	6-016066	BACATS	I*2	6-016032	BATØT	I*2	6-023352	BLANK	L*1	4-000170
BUCATS	I*2	6-01[illegible]	BUTØT	I*2	6-023354	CLADUM	I*2	7-000000	CLUDUM	I*2	7-000002	CNT	I*2	4-000452
DØ	I*2	6-014[illegible]	DU	I*2	6-014202	I	I*2	4-000440	J	I*2	4-000442	JJ	I*2	4-000454
K	I*2	4-000[illegible]	L	I*2	4-000446	LL	I*2	4-000454	M	I*2	4-000450	N	I*2	4-000460
NC	I*2	4-000[illegible]	NLABS	I*2	6-026706	NLINE	I*2	4-000456	NT	I*2	4-000436	ØNE	I*2	4-000430
PASS	I*2	F-000[illegible]*	PCCG1	R*4	6-016100	PCCG2	R*4	6-016104	PCC1	R*4	6-016070	PCC2	R*4	6-016074
PCLG1	R*4	6-016120	PCLG2	R*4	6-016124	PCL1	R*4	6-016110	PCL2	R*4	6-016114	SLASH	I*2	4-000432
TAPNØ	I*2	F-000[illegible]4*	TC	I*	6-014204	TWØ	I*2	4-000434						

ARRAYS

NAME	TYPE	ADDRESS	SIZE		DIMENSIØNS
A	R*4	6-014150	000030	12	(3,2)
BACAIN	R*4	6-00[illegible]	005220	1352	(26,26)
BACLAB	L*1	6-01[illegible]	000032	13	(26)
BACØR	I*2	6-00[illegible]	000074	30	(30)
BALABL	L*1	6-014[illegible]	000032	13	(26)
BAPØP	I*2	6-00[illegible]	000074	30	(30)
BARANS	R*4	6-01[illegible]	000170	60	(30)
BARØØN	R*4	6-02[illegible]	001000	256	(2,8,8)
BATØØN	R*4	6-02[illegible]	000100	32	(2,8)
BAUNØZ	I*2	6-00[illegible]	000074	30	(30)
BAVAR	R*4	6-01[illegible]	000170	60	(30)
BCVA	I*2	6-02[illegible]	002510	676	(26,26)
BCVB	I*2	6-02[illegible]	002510	676	(26,26)
BUCAIN	R*4	6-00[illegible]	005220	1352	(26,26)
BUCLAB	L*1	6-01[illegible]	000032	13	(26)
BUCØR	I*2	6-00[illegible]	000074	30	(30)

BULABL	L*1	6-[illegible]	[illegible]032	13	(26)
BU[illegible]P	I*2	6-[illegible]2	[illegible]074	3[illegible]	(30)
BURALS	R*4	6-[illegible]	[illegible]172	6[illegible]	(30)
[illegible]	R*4	5-[illegible]	[illegible]	2[illegible]	(2,[illegible],[illegible])
[illegible]	R*4	6-[illegible]	[illegible]	2	(2,8)
BU[illegible]	I*2	6-[illegible]	[illegible]074	30	(30)
BUVAR	R*4	6-[illegible]	[illegible]17[illegible]	6[illegible]	(30)
C[illegible]S	R*4	6-[illegible]	[illegible]	4	(2)
CLA[illegible]	R*4	6-[illegible]	[illegible]	4	(2)
CLASFY	L*1	5-[illegible]	[illegible]542	2[illegible]	(418)
CLO[illegible]T	L*1	8-[illegible]	[illegible]642	2[illegible]	(418)
CL[illegible]	R*4	6-[illegible]	[illegible]	4	(2)
C[illegible]	I*2	4-[illegible]2	[illegible]046	1[illegible]	(19)
GRAINS	L*1	6-[illegible]	[illegible]032	3	(25)
IBUF	L*1	F-[illegible]	[illegible]	[illegible]	(1)
LFLED	L*1	6-[illegible]	[illegible]321	1[illegible]4	(209)
PP	L*1	4-[illegible]	[illegible]17[illegible]	6[illegible]	(120)
PPATR	L*1	4-[illegible]17[illegible]	[illegible]170	6[illegible]	(120)
RC[illegible]S	R*4	6-[illegible]74	0010[illegible]	2[illegible]	(2,8,8)
SF[illegible]	L*1	7-[illegible]	[illegible]4	2	(4)
TC[illegible]S	R*4	6-[illegible]	[illegible]	32	(2,8)
TYPE	L*1	6-[illegible]	[illegible]321	1[illegible]4	(209)

LABELS

LABEL	ADDRESS	LABEL	ADDRESS	LABEL	ADDRESS	LABEL	ADDRESS	LABEL	ADDRESS
1	1-0[illegible]0054	2	1-[illegible]0630	3	**	5	1-001344	6	1-001216
7	1-0[illegible]1254	1[illegible]	**	12	**	20	**	59	**
96	**	97	**	98	**	99'	3-00245A	100'	3-000000
101'	3-[illegible]054	102'	3-[illegible]074	103'	3-000112	104'	3-00015[illegible]	105'	3-000206
106'	3-[illegible]0244	107'	3-[illegible]0304	108'	3-000344	109'	3-000376	110'	3-000424
111'	3-[illegible]	112'	3-[illegible]502	113'	3-000524	114'	**	115'	3-000610
116'	3-[illegible]0644	117'	3-[illegible]0720	118'	3-000752	119'	3-001002	120'	3-001026
121'	**	122'	3-[illegible]104	123'	3-001154	124'	3-00123[illegible]	125'	3-001244
126'	3-0[illegible]1270	127'	3-[illegible]1326	128'	3-001364	129'	3-001432	130'	3-00144[illegible]
132'	3-[illegible]	133'	**	134'	3-001506	135'	3-001A32	145'	3-001654
146'	3-[illegible]1716	147'	3-[illegible]1762	148'	3-002026	150'	3-002062	151'	3-002132
152'	3-[illegible]	153'	3-[illegible]2250	154'	3-00232[illegible]	155'	3-00235[illegible]	160	**
161	1-[illegible]6236	1[illegible]	**	190	**	196	1-005552	202	1-006274
205	1-[illegible]7[illegible]12	2[illegible]	1-[illegible]6660	207	1-006716	210	**	212	**
250	**	297	**	298	**	420	**	421	1-013206
45[illegible]	1-0[illegible]3525								

TOTAL SPACE ALLOCATED = 046732 9965

NO FPP INSTRUCTIONS GENERATED

BIAPRT,LP1=BIAPRT

```
              [illegible] TIME [illegible]
              I[illegible] IMIT I[illegible](A-Z)
              [illegible]
         *    BYTE [illegible]SFY(416),[illegible](416)
         *    [illegible]*1 TYPE(2[illegible]9),[illegible]BLED(2C9)
         *    [illegible]*1 [illegible](26),BACLAB(26),BUCLAB(26),BULABL(26),GRAINS(26)
         *    [illegible]I,[illegible],[illegible]G2,[illegible]G2,PCL1,PCL2,PCLG1,PCLG2
         *    [illegible]LAD,[illegible],RCZNS,CCZNS
         *    [illegible]
         *    DIM[illegible](2),[illegible](2),T[illegible]S(2,8),CC[illegible]S(2),RCZNS(2,8,8)
         *    [illegible](2,8),[illegible](2,8),BARC[illegible](2,8,8),BURC[illegible](2,8,8)
         *    [illegible](2,[illegible]),[illegible](26,26)
         *    [illegible]
         *    [illegible]/CHI[illegible]/[illegible],BAC[illegible],BU[illegible]CZ,BACZR,BUCZR,BACAIN,BUCAIN,
         *         *     [illegible]AT,BARANS,BURANS,A,[illegible],[illegible],TC,BUCATS,BUCLAB,
         *         *     [illegible]LABL,CLASFY,TYPE,L[illegible]ED,BACATS,BACLAB,A[illegible]CZ,
         *         *     [illegible]1,[illegible]2,PCCG1,PCCG2,PCL1,PCL2,PCLG1,PCLG2,APZP,
         *         *     [illegible]A,[illegible],RATZT,[illegible]LTZT,CLAD,CLUP,TC[illegible]S,RCZNS,CCZNS,
         *         *     [illegible],BARCZN,BURCZN,[illegible]LABS,GRAINS
         *    [illegible]/DUM[illegible]/[illegible],[illegible]
         *    [illegible]/[illegible]
         *    [illegible](3,2)
         *    [illegible](30),[illegible](30),BACZR(30),BUCZR(30)
         *    [illegible](26,26),BAVAR(30),BUVAR(30)
         *    [illegible]BARANS(30),[illegible]S(30)
              I[illegible]=[illegible]
              [illegible]
              D[illegible] I=1,26
      5       GRA[illegible]S(I)=[illegible]
              D[illegible] I=1,2
              RE[illegible],101) [illegible]S(I)
     101      F[illegible](F10.5)
              RE[illegible],10[illegible]) (T[illegible]S(I,J),J=1,8)
              D[illegible]=1,8
              RE[illegible],10[illegible]) (RC[illegible]S(I,J,K),K=1,8)
      9       C[illegible]E
      1[illegible]     C[illegible]E
     10[illegible]     F[illegible](8F1[illegible],6)
C 50          TYP[illegible]
C56           FORMAT(1H ,' I[illegible]T [illegible]HBZ ')
C             ACC[illegible]55,GR[illegible]S
      50      REA[illegible],55)GRAINS
      55      FORMAT(26A1)
              [illegible] I=1,26
              IF(GRAINS(I).EQ.' ') GOTO 65
      60      NLA[illegible]=NLABS+1
      65      IF([illegible].GT.[illegible]) RETURN
              GRAINS(1)='S'
              GRA[illegible](2)='[illegible]'
              GRA[illegible](3)='-'
              [illegible]=[illegible]
              RET[illegible]
              END
```

PROGRAM SECTIONS

NUMBER	NAME	SIZE		ATTRIBUTES
1	?CODE1	[illegible]	134	RW,I,CON,LCL
3	?IDATA	[illegible]	11	RW,D,CON,LCL
4	?VARS	[illegible]	1	RW,D,CON,LCL
5	DBIAS	?2????	5873	RW,D,OVR,GBL
7	??MMY	[illegible]	2	RW,D,OVR,GBL
8	?L?T??	???4?2	2?9	RW,D,OVR,GBL

ENTRY POINTS

NAME	TYPE	ADDRESS	NAME	TYPE	ADDRESS	NAME	TYPE	ADDRESS	NAME	TYPE	ADDRESS	NAME	TYPE	ADDRESS
C??HE?		1-000??0												

VARIABLES

NAME	TYPE	ADDRESS	NAME	TYPE	ADDRESS	NAME	TYPE	ADDRESS	NAME	TYPE	ADDRESS	NAME	TYPE	ADDRESS
AP?P	I*2	6-?1??30	A????	I*2	6-016060	BACATS	I*2	6-016032	BATZT	I*2	6-023352	BUCATS	I*2	6-014236
B?T?T	I*?	6-023?54	CLA???	I*?	7-000000	CLUDUM	I*2	7-000002	DA	I*2	6-014230	DU	I*2	6-014272
I	I*2	4-00??02	I?	I*2	4-000000	J	I*2	4-000004	K	I*2	4-000006	NLARG	I*2	6-026706
PCOG1	R*4	6-01????	PCOG2	R*4	6-016104	PCC1	R*4	6-016070	PCC2	R*4	6-016074	PCLG1	R*4	6-016120
PCLG2	R*4	6-01??74	PCL1	R*4	6-016110	PCL2	R*4	6-016114	TC	I*2	6-014264			

ARRAYS

NAME	TYPE	ADDRESS	SIZE		DIMENSIONS
A	R*4	6-0141??	000030	12	(3,2)
BACAIN	R*4	6-??????	??5270	?372	(26,26)
BACLA?	L*1	6-01??34	00??3?	?3	(26)
BACPR	I*2	6-??????	???074	[illegible]	(30)
BALABL	L*1	6-014?74	00??32	13	(26)
BAPTP	I*2	6-0?????	???074	30	(30)
BARANS	R*4	6-013???	00?17?	6?	(30)
BAPC??	R*4	6-02?7?6	?01???	256	(2,2,8)
BATC??	R*4	6-024??6	??0??	?	(2,8)
BAU?C?	I*2	6-??????	???074	3?	(30)
BAVAR	R*4	6-013???	??0170	6?	(30)
BCVA	I*2	6-?1??3?	??251?	67?	(26,26)
BCVU	I*2	6-02????	??251?	676	(26,26)
BUCAIN	R*4	6-0?????	??5270	?372	(26,26)
BUCLAR	L*1	6-01????	????32	13	(26)
BUCPR	I*2	6-??????	???074	[illegible]	(30)
BULABL	L*1	6-01????	????32	3	(26)
BUPTP	I*2	6-????74	???074	3?	(30)
BURANS	R*4	6-01????	??0170	6?	(30)
BUPC??	R*4	6-02????	?01???	2?6	(2,8,8)
BUTC??	R*4	6-??????	???1??	3?	(2,8)
BUU?C?	I*2	6-??????	???074	[illegible]	(30)

[illegible]VAR	R*4	6-[illegible]	[illegible]170	6[illegible]	(30)
[illegible]S	R*4	5-[illegible]	[illegible]	4	(2)
CLA[illegible]	R*4	6-[illegible]	[illegible]	4	(2)
CLASFY	L*1	5-[illegible]	[illegible]32	[illegible]9	(418)
CL[illegible]T	L*1	6-[illegible]	[illegible]642	[illegible]9	(418)
[illegible]	R*4	5-[illegible]	[illegible]	4	(2)
[illegible]S	L*1	6-[illegible]	[illegible]	13	(26)
[illegible]	L*1	5-[illegible]	[illegible]	[illegible]	(2[illegible]9)
[illegible]S	R*4	6-[illegible]	[illegible]	256	(2,8,8)
T[illegible]S	R*4	6-[illegible]	[illegible]	32	(2,8)
TYPE	L*1	6-[illegible]	[illegible]	1[illegible]4	(209)

LABELS

LABEL	ADDRESS	LABEL	ADDRESS	LABEL	ADDRESS	LABEL	ADDRESS	LABEL	ADDRESS
5	**	9	**	10	**	50	**	55'	3-000012
60	**	65	1-[illegible]0510	100'	3-000004	101'	3-000000		

TOTAL SPACE ALLOCATED = 033426 [illegible]

NO FPP INSTRUCTIONS GENERATED

[illegible]

```
0153              IF(TRUF(1,N).EQ.CAT(JJ)) GØ TØ 270
0154     260      CØNTINUE
0155              NC=NC+1
0156              CAT(NC)=TRUF(1,N)
0157     270      CØNTINUE
                  C NØW SØRT THE ENTRIES BY DØT DISTANCE
0158     271      DØ 280 II=1,N
0159              T=9999
0160              DØ 275 JJ=1,N
0161              IF(T.LT.DRUF(JJ)) GØ TØ 275
0162              PTAR(II)=JJ
0163              T=DRUF(JJ)
0164     275      CØNTINUE
0165              JK=PTAR(II)
0166              DRUF(JK)=9999
0167     280      CØNTINUE
0168     44       CØNTINUE
0169       105    FØRMAT(I5)
0170              XL2=L2DIST(J)
0171              XL2=XL2/100.
0172     210      FØRMAT(I2)
0173              WRITE(3,201)
0174     201      FØRMAT('0',70X,'BRIGHTNESS  GREEN NUMBER')
0175     202      FØRMAT(I4)
0176              WRITE(3,203) (CLASSE(L,J),L=1,6),A1B(J),A1G(J)
0177     203      FØRMAT(' ',16X,'CLUSTER NAME: ',2X,6A1,18X,'ACQUISITIØN 1',6
                2I4,8X,I3)
0178              WRITE(3,211) (SCLASS(L,J),L=1,6),A2B(J),A2G(J)
0179     211      FØRMAT(' ',16X,'LABELING DØT: ',2X,6A1,18X,'ACQUISITIØN 2',6
                2I4,8X,I3)
0180              WRITE(3,212) XL2,A3B(J),A3G(J)
0181     212      FØRMAT(' ',16X,'L2 DISTANCE: ',2X,F6.2,18X,'ACQUISITIØN 3',
                2I4,8X,I3)
0182              WRITE(3,213) CAT,A4B(J),A4G(J)
0183     213      FØRMAT(' ',16X,'CATAGØRIES:  ',2X,8(X,A1),8X,'ACQUISITIØN 4
                2I4,8X,I3)
0184              WRITE(3,99)
                  C
                  C   NØW ØUTPUT DØTS IF ANY
0185              IF(N.EQ.0) GØ TØ 106
0186              WRITE(3,214)
0187     214      FØRMAT(' ',10X,'DØTS    DISTANCE    DØTS    DISTANCE    D
                2'    DISTANCE    DØTS    DISTANCE    DØTS    DISTANCE')
0188              WRITE(3,99)
0189              DØ 216 II=1,N,5
0190              FF=II+4
0191              LIM=5
0192              IF(FF.LE.N) GØ TØ 220
0193              FF=N
0194              LIM=N-II+1
0195     220      LL=1
0196              DØ 217 JJ=II,FF
0197              NDX=PTAR(JJ)
0198              DØ 218 KK=1,5
0199              DD(KK,LL)=TRUF(KK,NDX)
0200     218      CØNTINUE
```

```
0201                DIS(LL)=DSBUF(NDX)
0202                LIS(LL)=DIS(LL)/1[illegible].
0203                LL=LL+1
0204          217   CONTINUE
0205                WRITE(3,215) ((DI(KK,JJ),KK=1,5),DIS(JJ),JJ=1,LIM)
0206          216   CONTINUE
0207          215   FORMAT(' ',9X,5(A1,1X,3A1,1/,A1,3),F6.2,5X))
                    C
                    C TEST FOR END OF DATA
                    C
0208          104   IF(J.EQ.ALSETS) [illegible] TO 4
0209                J=J+1
0210                IF(K.EQ.MAXC) GO TO 15
0211                K=K+1
0212                GO TO 2
                    C     FINISHED 1 PAGE, GO TOP OF FORM AND HEADER
0213          15    WRITE(3,98)
0214                WRITE(3,100)
0215                MAXC=5
0216                GO TO 1
                    C
0217           4    CONTINUE
0218                WRITE(3,50)
0219          50    FORMAT('0',10X,'CLUSTERING CHANNEL LIST:')
0220                WRITE(3,51)  (CT(I),I=1,CHMAX)
0221          51    FORMAT(' ',10X,16(1X,I2))
0222                RETURN
0223                END
```

PROGRAM SECTIONS

NUMBER	NAME	SIZE		ATTRIBUTES
1	$CODE1	005076	1311	RW,I,CON,LCL
3	$IDATA	001214	326	RW,D,CON,LCL
4	$VARS	011464	2458	RW,D,CON,LCL
5	$TEMPS	000012	5	RW,D,CON,LCL
6	CBIAS	005122	1321	RW,D,OVR,GBL
7	DUMMY	000004	2	RW,D,OVR,GBL
8	CLCOM	000642	209	RW,D,OVR,GBL

ENTRY POINTS

NAME	TYPE	ADDRESS	NAME	TYPE	ADDRESS	NAME	TYPE	ADDRESS	NAME	TYPE	ADDRESS	NAME	TYPE	ADDRESS
CLURES		1-000000												

VARIABLES

NAME	TYPE	ADDRESS	NAME	TYPE	ADDRESS	NAME	TYPE	ADDRESS	NAME	TYPE	ADDRESS	NAME	TYPE	ADDRESS
ALSETS	I*2	4-011376	BACATS	I*2	6-003314	BATST	I*2	6-003346	BLANK	I*2	4-011376	BOCATS	I*2	6-001556
BUTST	I*2	6-003370	CURAX	I*2	4-011406	CLABUM	I*2	7-000000	CLUBUM	I*2	7-000002	D	I*2	4-011442
DFLD	I*2	4-011416	DP	I*2	6-001550	DRPT	I*2	4-011410	DU	I*2	6-001552	EF	I*2	4-011454
FILE	I*2	F-000006*	I	I*2	4-011402	II	I*2	4-011434	IJ	I*2	4-011432	J	I*2	4-011404
JJ	I*2	4-011426	JK	I*2	4-011450	K	I*2	4-011420	KK	I*2	4-011442	L	I*2	4-011452
LFLD	I*2	4-011414	LIM	I*2	4-011456	LL	I*2	4-011440	MAXC	I*2	4-011430	N	I*2	4-011440
NC	I*2	4-011444	NDX	I*2	4-011436	NUM	I*2	4-011424	N1	L*1	4-003536	N2	L*1	4-006307
PCCG1	R*4	6-003336	PCCG2	R*4	6-003342	PCC1	R*4	6-003326	PCC2	R*4	6-003332	PCLG1	R*4	6-003356
PCLG2	R*4	6-003362	PCL1	R*4	6-003346	PCL2	R*4	6-003352	PT	I*2	4-011422	R	I*2	F-000004*
RJ	I*2	4-011412	SETSR	I*2	4-011403	T	I*2	4-011446	TC	I*2	6-001554	X8	R*4	4-000004
XG	R*4	4-000010	XL2	R*4	4-000000									

ARRAYS

NAME	TYPE	ADDRESS	SIZE		DIMENSIONS
A	R*4	6-001520	000030	12	(3,2)
A1B	I*2	4-007400	000170	60	(60)
A1G	I*2	4-007210	000170	60	(60)
A2B	I*2	4-007760	000170	60	(60)
A2G	I*2	4-007570	000170	60	(60)
A3B	I*2	4-010340	000170	60	(60)
A3G	I*2	4-010150	000170	60	(60)
A4B	I*2	4-010720	000170	60	(60)
A4G	I*2	4-010530	000170	60	(60)
BACAIN	R*4	6-000220	001400	128	(8,8)
BACLAB	L*1	6-003316	000010	4	(8)
BACOR	I*2	6-000140	000030	12	(12)
BALABL	L*1	6-001600	000010	4	(8)
BAPOP	I*2	6-000030	000030	12	(12)
BARANS	R*4	6-001360	000060	24	(12)

```
CLURES.FTN      /TR:BLOCKS/WR
BAUNCP  I*2   6-000060   000030    12    (12)
BAVAR   R*4   6-001220   000060    24    (12)
BCVA    I*2   6-003356   000200    64    (8,8)
BCVB    I*2   6-003564   000200    64    (8,8)
BUCAID  R*4   6-000620   000400   128    (8,8)
BUCLAS  L*1   6-001560   000010     4    (8)
BUCNR   I*2   6-000170   000030    12    (12)
BULABL  L*1   6-001570   000010     4    (8)
BURCP   I*2   6-000030   000030    12    (12)
BURANS  R*4   6-001440   000060    24    (12)
BUUNCR  I*2   6-000110   000030    12    (12)
BUVAR   R*4   6-001300   000060    24    (12)
CAT     L*1   4-000040   000010     4    (8)
CORNS   R*4   6-005112   000010     4    (2)
CDNAME  I*2   4-006724   000170    60    (60)
CLAP    R*4   6-003772   000010     4    (2)
CLASFY  L*1   6-001610   000642   209    (418)
CLSCT   L*1   3-000000   000642   209    (418)
CLUD    R*4   6-004002   000010     4    (2)
CNAME   L*1   4-006050   000553   150    (6,60)
CT      I*2   4-006570   000040    16    (16)
DRUF    I*2   4-007114   000074    30    (30)
DD      L*1   4-004536   000071    12    (5,5)
DIS     R*4   4-000024   000024    10    (5)
DTBUF   L*1   4-001370   004716  2255    (2515)
DSBUF   I*2   4-011300   000074    30    (30)
IBUF    L*1   F-000002*  000001     0    (1)
LBLED   L*1   6-002773   000321   104    (209)
L2DIST  I*2   4-011110   000170    60    (60)
PTAS    I*2   4-006630   000074    30    (30)
RC2NS   R*4   6-004112   001600   256    (2,8,8)
SCLASS  L*1   4-000620   000550   180    (6,60)
TRUF    L*1   4-006310   000226    75    (5,30)
TC2NS   R*4   6-004012   000100    32    (2,8)
TYPE    L*1   6-002452   000321   104    (209)
*
*
LABELS
*
LABEL   ADDRESS    LABEL   ADDRESS    LABEL   ADDRESS    LABEL   ADDRESS    LABEL   ADDRESS
*
1       1-002452   2       1-002466   4       1-004770   10      1-000250   12      1-010722
15      1-004713   19      1-000434   20      **         21      **         37      1-001474
39      **         40      **         41      **         42      **         43      1-002166
44      1-003432   45      **         46      1-002414   47      1-002360   50'     3-001104
51'     3-001146   98'     3-000004   99'     3-000040   100'    3-000062   102'    3-000000
103'    3-000306   105'    3-000312   106     1-004644   110'    3-000044   200'    3-000134
201'    3-000316   202'    3-000356   203'    3-000360   204'    3-000000   210'    **
211'    3-000444   212'    3-000526   213'    3-000620   214'    3-000674   215'    3-001754
216     **         217     **         218     **         223     1-004254   249     1-002770
250     **         260     **         270     1-003200   271     1-003222   275     1-003342
280     **         1000'   3-000130   1001'   3-000170   1002'   3-000210   1003'   3-000240
*
*
FUNCTIONS AND SUBROUTINES REFERENCED
*
```

```
FØRTRAN IV-PLUS V02-51          [illegible]02154    09-MAR-78          PAGE 8
CLURES.FTN      /TR:BLØCKS/WP
 CDRED
@
@
TØTAL SPACE ALLØCATED = 026000  5632
@
CLURES,LP1=CLURES
```

```
STDATA.FTN        /TR:BLOCKS/WR
      SUBROUTINE STDATA(IBUF,R,FILE,DFLG)
      IMPLICIT INTEGER(A-Z)
      INCLUDE 'COMMON1.PAT'
      BYTE CLASFY(418),CLDET(418)
       LOGICAL*1 TYPE(209),LBLED(209)
       LOGICAL*1 BALABL(8),BACLAB(8),BUCLAB(8),BULABL(8)
      REAL PCC1,PCC2,PCCG1,PCCG2,PCL1,PCL2,PCLG1,PCLG2
       REAL CLAD,CLUD,TCONS,RCONS,CCONS
      DIMENSION CLAD(2),CLUD(2),TCONS(2,8),CCONS(2),RCONS(2,8,8)
       DIMENSION BCVA(8,5),BCVU(8,8)
       DIMENSION PARZP(12),SUPZP(12)
       COMMON/CBIAS/BAPZP,BUPZP,BAUNCO,BUUNCO,BACOR,BUCOR,BACAIN,BUCAIN,
     *           BAVAR,BUVAR,BARANS,BURANS,A,DA,DU,TC,BUCLTS,BUCLAB,
     *           BULABL,BALABL,CLASFY,TYPE,LBLED,BACATS,BACLAB,
     *           PCC1,PCC2,PCCG1,PCCG2,PCL1,PCL2,PCLG1,PCLG2,
     *           BCVA,BCVU,PATZT,BLTZT,CLAD,CLUD,TCONS,RCONS,CCONS
       COMMON/DUMMY/CLADUM,CLUDUM
      COMMON/CLCOM/CLDET
       REAL A(3,2)
       DIMENSION BAUNCO(12),BUUNCO(12),BACOR(12),BUCOR(12)
       REAL BACAIN(8,8),BUCAIN(8,8),BAVAR(12),BUVAR(12),BARANS(12),BURANS(12)
      LOGICAL*1 IBUF(1),PEN(132),PP(132),PN(132)
      LOGICAL*1 PTL(132),PK(2,132)
      LOGICAL*1 BLANK
       LOGICAL*1 PF(132)
      REAL T1
       COMMON/PCNT/LINE,PAGE
      DATA BLANK/1H /
       IF(DFLG.EQ.0)  GO TO 12
       CALL STDMP(IBUF)
       GO TO 13
 12    CONTINUE
 13    CONTINUE
       NMAX=110
      DPZP=0
      K=1
      I=1
      PI=11
      DECODE(3,99,IBUF(5))  ALSETS
       IF(ALSETS.EQ.0)  RETURN
 99   FORMAT(I3)
      DECODE(2,98,IBUF(10))  NCH
 98   FORMAT(I2)
      LINE=66
      CALL BNT
       IF(IBUF(2).EQ.'S')  LINE=LINE+2
       NLINS=59-LINE
       LSAV=LINE
       XL=NCH+6
       IF(DFLG.EQ.1) NLINS=NLINS-7
       LSETS=NLINS/XL
       IF(LSETS.GT.0) WRITE(3,100)
 100   FORMAT('0 ')
       IF(LSETS.GT.0)  GO TO 8
       LINE=66
       CALL BNT
```

ORIGINAL PAGE IS OF POOR QUALITY

```
STDATA,FTN      /TRIBLØCKS/WP
0053            LSETS=60/XL
0054    8       CØNTINUE
0055            IF(IBUF(2).EQ.'F')  GØ TØ 16
0056            WRITE(3,97)
0057      97    FØRMAT(' ',49X,'STATISTICS REPØRT')
0058      16    CØNTINUE
0059            ØSETS=5
0060            PSETS=0
0061            LFLD=11+9*NCH
0062            IF(IBUF(2).EQ.'F')  LFLD=18+9*NCH
0063    20      J=1
0064            RJ=12
0065            IF(K.EQ.1)  GØ TØ 6
0066            CALL CØRED(IBUF,R,FILE)
0067    4       DECØDE(2,98,IBUF(R))  SETSR
0068             IF(SETSR.EQ.0)  RETURN
0069    1        CØNTINUE
0070            IF(IBUF(2).EQ.'S')  GØ TØ 22
0071            CALL FNAMF(IBUF(RJ),PFN(PI))
0072    22      CALL PØP(IBUF(RJ),PP(PI),IBUF(2))
0073            CALL SNAMF(IBUF(RJ),PN(PI),IBUF(2))
0074            CALL MØTTL(PTL(PI))
0075             CALL MEAN(IBUF(RJ),PM(PI),NCH,IBUF(2))
0076            DECØDE(5,96,IBUF(RJ+6)) DUM1
0077    96       FØRMAT(I5)
0078            DPØP=DPØP+DUM1
0079            IF(K.EQ.ALSETS)  GØ TØ 10
0080            K=K+1
0081            IF(I.EQ.ØSETS)  GØ TØ 9
0082            I=I+1
0083            PI=PI+19
0084    2       IF(J.EQ.SETSR)  GØ TØ 20
0085            J=J+1
0086            RJ=RJ+LFLD
0087            GØ TØ 1
0088    9        K=K-1
0089    10      CØNTINUE
0090             PSETS=PSETS+1
0091             IF(PSETS.LE.LSETS)  GØ TØ 11
0092             PSETS=1
0093             IF(DFLG.EQ.1)  GØ TØ 14
0094             LINE=66
0095             CALL BNT
0096             LSETS=60/XL
0097             GØ TØ 11
0098    14       CØNTINUE
0099             LINE=66
0100             CALL KNT
0101    11       CØNTINUE
0102             WRITE(3,90)
0103    90       FØRMAT(1HØ,' ')
0104            IF(IBUF(2).EQ.'S')  GØ TØ 3
0105            WRITE(3,101)  (PFN(N),N=1,NMAX)
0106    101     FØRMAT(1H ,131A1)
0107            WRITE(3,102)  (PP(N),N=10,NMAX)
0108    102     FØRMAT(1H ,'CHANNEL ',122A1)
```

```
0109                WRITE(3,101)  (PN(N),N=1,NMAX)
0110                GØ TØ 7
0111      3          CØNTINUE
0112                WRITE(3,101)  (PN(N),N=1,NMAX)
0113                WRITE(3,102)  (PP(N),N=9,NMAX)
0114      7         WRITE(3,103)  (PTL(N),N=8,NMAX)
0115      103       FØRMAT(1H ,'NUMBER ',122A1)
0116                IN=6
0117                IM=NMAX
0118                DØ 5  II=1,NCH
0119                WRITE(3,104)  (II,(PM(N),N=IN,IM))
0120      104       FØRMAT(1H ,2X,I2,128A1)
0121                IN=IN+132
0122                IM=IM+132
0123      5         CØNTINUE
0124      6         CØNTINUE
0125                DØ  30 N=1,132
0126                PF(N)=BLANK
0127                 PFN(N)=BLANK
0128                PN(N)=BLANK
0129                PTL(N)=BLANK
0130                PP(N)=BLANK
0131                DØ  40  NN=1,NCH
0132                PM(N+(NN-1)*132)=BLANK
0133      40        CØNTINUE
0134      30        CØNTINUE
0135                IF(K.EQ.1)  GØ TØ 4
0136                IF(K.EQ.ALSETS)  GØ TØ 15
0137                 K=K+1
0138                I=1
0139                PI=11
0140                GØ TØ 2
0141      15         CØNTINUE
0142                 IF(LINE.EQ.0) LSAV=0
0143                 LINE=PSETS*XL+LSAV
0144                 CALL BNT
0145                BUPØP(10)=22932-DPØP
0146                BUUNCØ(10)=((22932.-DPØP)/22932.)*100
0147                 RETURN
0148                END
```

STDATA,FTN /TR:BL2CKS/WR

PRØGRAM SECTIØNS

NUMBER	NAME	SIZE		ATTRIBUTES
1	$CØDE1	003012	773	RW,I,CØN,LCL
2	$PDATA	000004	2	RW,D,CØN,LCL
3	$IDATA	000210	68	RW,D,CØN,LCL
4	$VARS	005410	1412	RW,D,CØN,LCL
5	$TEMPS	000006	3	RW,D,CØN,LCL
6	CBIAS	005122	1321	RW,D,ØVR,GBL
7	DUMMY	000004	2	RW,D,ØVR,GBL
8	CLCØM	000642	209	RW,D,ØVR,GBL
9	PCNT	000004	2	RW,D,ØVR,GBL

ENTRY PØINTS

NAME	TYPE	ADDRESS
STDATA		1-000000

VARIABLES

NAME	TYPE	ADDRESS	NAME	TYPE	ADDRESS	NAME	TYPE	ADDRESS	NAME	TYPE	ADDRESS	NAME	TYPE	ADDRESS
ALSETS	I*2	4-005344	BACATS	I*2	6-003314	BATØT	I*2	6-003766	BLANK	L*1	4-005190	BUCATS	I*2	6-001556
BUTØT	I*2	6-003770	CLADUM	I*2	7-000000	CLUDUM	I*2	7-000002	DFLG	I*2	F-000010*	DØ	I*2	6-001550
DPØP	I*2	4-005334	DU	I*2	6-001552	DUM1	I*2	4-005374	FILE	I*2	F-000006*	I	I*2	4-005340
II	I*2	4-005404	IM	I*2	4-005402	IN	I*2	4-005400	J	I*2	4-005366	K	I*2	4-005336
LFLD	I*2	4-005364	LINE	I*2	9-000000	LSAV	I*2	4-005352	LSETS	I*2	4-005356	N	I*2	4-005376
NCH	I*2	4-005346	NLINS	I*2	4-005350	NMAX	I*2	4-005332	NN	I*2	4-005406	ØSETS	I*2	4-005360
PAGE	I*2	9-000002	PCCG1	R*4	6-003336	PCCG2	R*4	6-003342	PCC1	R*4	6-003326	PCC2	R*4	6-003332
PCLG1	R*4	6-003356	PCLG2	R*4	6-003362	PCL1	R*4	6-003346	PCL2	R*4	6-003352	PI	I*2	4-005342
PSETS	I*2	4-005362	R	I*2	F-000004*	RJ	I*2	4-005370	SETSR	I*2	4-005372	TC	I*2	6-001554
T1	R*4	4-005326	XL	I*2	4-005354									

ARRAYS

NAME	TYPE	ADDRESS	SIZE		DIMENSIØNS
A	R*4	6-001520	000030	12	(3,2)
BACAIN	R*4	6-000220	000400	128	(8,8)
BACLAB	L*1	6-003316	000010	4	(8)
BACØR	I*2	6-000140	000030	12	(12)
BALABL	L*1	6-001600	000010	4	(8)
BAPØP	I*2	6-000000	000030	12	(12)
BARANS	R*4	6-001350	000060	24	(12)
BAUNCØ	I*2	6-000060	000030	12	(12)
BAVAR	R*4	6-001220	000060	24	(12)
BCVA	I*2	6-003366	000200	64	(8,8)
BCVU	I*2	6-003566	000200	64	(8,8)
BUCAIN	R*4	6-000620	000400	128	(8,8)
BUCLAB	L*1	6-001560	000010	4	(8)
BUCØR	I*2	6-000170	000030	12	(12)

BULABL	L*1	6-001570	000010	4	(8)
BUPØP	I*2	6-000030	000030	12	(12)
BURANS	R*4	6-001440	000060	24	(12)
BUUNCØ	I*2	6-000110	000030	12	(12)
BUVAR	R*4	6-001300	000060	24	(12)
CCØNS	R*4	6-005112	000010	4	(2)
CLAD	R*4	6-003772	000010	4	(2)
CLASFY	L*1	6-001610	000642	209	(418)
CLDØT	L*1	8-000000	000642	209	(418)
CLUD	R*4	6-004002	000010	4	(2)
IBUF	L*1	F-000002*	000001	0	(1)
LBLED	L*1	6-002773	000321	104	(209)
PF	L*1	4-005121	000204	66	(132)
PFN	L*1	4-000000	000204	66	(132)
PM	L*1	4-001020	004100	1056	(2112)
PN	L*1	4-000410	000204	66	(132)
PP	L*1	4-000204	000204	66	(132)
PTL	L*1	4-000614	000204	66	(132)
RCØNS	R*4	6-004112	001000	256	(2,8,8)
TCØNS	R*4	6-004012	000100	32	(2,8)
TYPE	L*1	6-002452	000321	104	(209)

LABELS

LABEL	ADDRESS	LABEL	ADDRESS	LABEL	ADDRESS	LABEL	ADDRESS	LABEL	ADDRESS
1	1-001004	2	1-001426	3	1-002104	4	1-000724	5	**
6	1-002466	7	1-002240	8	1-000500	9	1-001464	10	1-001500
11	1-001632	12	1-000072	13	1-000100	14	1-001502	15	1-002666
16	1-000542	20	1-000630	22	1-001072	30	**	40	**
90'	3-000052	93'	3-000046	97'	3-000016	98'	3-000004	99'	3-000000
100'	3-000010	101'	3-000060	102'	3-000066	103'	3-000106	104'	3-000126

FUNCTIØNS AND SUBRØUTINES REFERENCED

BNT CØRED FNAME KNT MØTTL MEAN PØP SNAME STØMP

TØTAL SPACE ALLØCATED = 016640 3792

STDATA,LP:=STDATA

```
      SUBROUTINE DOTRAY(IBUF,A,FILE,PRTOUT)
      IMPLICIT INTEGER (A-Z)
      BYTE IBUF(1)
      INCLUDE 'COMMON1.DAT'
 *    BYTE CLASFY(418),CLDST(418)
 *    LOGICAL*1 TYPE(209),LABEL(209)
 *    LOGICAL*1 BALABL(26),BACLAB(26),BUCLAB(26),BULABL(26),GRAINS(26)
 *    REAL PCC1,PCC2,PCCS1,PCCS2,PCL1,PCL2,PCLS1,PCLS2
 *    REAL CLAD,CLUD,TCONS,RCONS,CCONS
 *    REAL RATCON,BUTCON,BARCON,BURCON
 *    DIMENSION CLAD(2),CLUD(2),TCONS(2,8),CCONS(2),RCONS(2,6,8)
 *    DIMENSION RATCON(2,8),BUTCON(2,8),BARCON(2,8,8),BURCON(2,8,8)
 *    DIMENSION PCVA(26,26),PCVL(26,26)
 *    DIMENSION BARSP(30),BURSP(30)
 *    COMMON /COMIAS/BARSP,BURSP,BALACP,BUUNCO,BACOR,BUCOR,BACAIN,BUCAIN,
 *   *     BAVAR,BUVAR,BARANS,BURANS,ADJT,[illegible],TC,BUCATS,BUCLAB,
 *   *     BULABL,BALABL,CLASFY,TYPE,LABEL,BACATS,BACLAB,AUNCO,
 *   *     PCC1,PCC2,PCCS1,PCCS2,PCL1,PCL2,PCLS1,PCLS2,APOP,
 *   *     PCVA,PCVL,RATZT,BUTZT,CLAD,CLUD,TCONS,RCONS,CCONS,
 *   *     RATCON,BUTCON,BARCON,BURCON,NVARS,GRAINS
 *    COMMON /DUMMY/CLADUM,CLUDUM
 *    COMMON /LCV2/CLDST
 *    REAL A(3,2)
 *    DIMENSION BATORL(30),BUTORL(30),BARIR(30),BURIR(30)
 *    REAL BACAIN(26,26),BUCAIN(26,26),BAVAR(30),BUVAR(30)
 *    REAL BARANS(30),BURANS(30)
      COMMON /PONT/LINE,PAGE
      REAL S(6),VC(2)
      INTEGER PAGE,FILE,R,REC,RC,SETSN,PRTOUT
      INTEGER I(5),IS(8),[illegible](19),PASS
      BYTE BLK,ONE,ZNBT,SLASH,TWO,LABEL(5)
      BYTE IC,CT(4)
      DATA ONE/'1'/,SLASH/'/'/,TWO/'2'/
      DATA CT/' ','L','M','S'/
      DATA THS/'T'/,THR/'R'/
      DATA BLK/' '/
      LFLD=44
      K=1
      REC=1
      WRITE(3,100)
      WRITE(3,104) (I,I=1,4)
      WRITE(3,101)
      WRITE(3,102)
      WRITE(3,103)
      WRITE(3,105)
      LCNT=4
      SETSN=15
      CONTINUE
      DECODE(2,200,IBUF(5))SLC1
      DECODE(2,200,IBUF(8))SLC2
      DECODE(2,200,IBUF(11))SLC3
      DECODE(2,200,IBUF(14))SLC4
      IF(IBUF(4).EQ.'-')SLC1=-SLC1
      IF(IBUF(7).EQ.'-')SLC2=-SLC2
      IF(IBUF(10).EQ.'-')SLC3=-SLC3
      IF(IBUF(13).EQ.'-')SLC4=-SLC4
```

```
              RJ=1
              N=1
      1       CONTINUE
              TYPE(K)=IBUF(RJ+12)
              LABEL(K)=IBUF(RJ+11)
              IF(IBUF(RJ+13) .EQ. '1') GO TO 8
              CLASFY(2*K)=IBUF(RJ+10)
              CLASFY(2*K-1)=IBUF(RJ+9)
              GO TO 9
      8       CLASFY(2*K)=TH2
              CLASFY(2*K-1)=TH1
      9      IF(LCNT.LT.60) GO TO 10
              WRITE(3,100)
              WRITE(3,106) (I,I=1,4)
              WRITE(3,101)
              WRITE(3,102)
              WRITE(3,103)
              WRITE(3,105)
             LCNT=5
      10      DECODE(1,200,TYPE(K)) ID
              ID=ID+1
              DECODE(3,300,IBUF(RJ+29))GRN1
              DECODE(3,300,IBUF(RJ+33))GRN2
              DECODE(3,300,IBUF(RJ+37))GRN3
              DECODE(3,300,IBUF(RJ+41))GRN4
              IF(IBUF(RJ+28).EQ.'-')GRN1=0-GRN1
              IF(IBUF(RJ+32).EQ.'-')GRN2=0-GRN2
              IF(IBUF(RJ+36).EQ.'-')GRN3=0-GRN3
              IF(IBUF(RJ+40).EQ.'-')GRN4=0-GRN4
              GRN1=GRN1-SL01
              GRN2=GRN2-SL02
              GRN3=GRN3-SL03
              GRN4=GRN4-SL04
              WRITE(3,104) (IBUF(RJ+L-1),L=1,9),PT(ID),LPLFD(K),CLDET(2*K-1),
             1CLDET(2*K),CLASFY(2*K-1),CLASFY(2*K),(IBUF(RJ+15+M),M=1,3),
             2 GRN1,(IBUF(RJ+19+M),M=1,3),
             3 GRN2,(IBUF(RJ+23+M),M=1,3),
             4 GRN3,(IBUF(RJ+2?+M),M=1,3), GRN4
              LCNT=LCNT+1
              K=K+1
              IF(N.EQ.SETSP) GO TO 13
              N=N+1
              RJ=RJ+LFIL
              GO TO 1
      13      IF(REC.EQ.14) GO TO 3
              CALL CDRED(IBUF,9,FILE)
              REC=REC+1
              IF(REC.EQ.14) SETSP=14
              GO TO 2
      3       PASS=2
              CALL CPL
              CALL HIAPRT(IBUF,TAB,?,PASS)
             IF(PRTOUT.EQ.1) RETURN
              IF(CLADUM.EQ.1) GO TO 20
             IF((PRT?.LT.?).OR.(?.LT.?)) PRTOUT=2
      20      IF(CLADUM.EQ.1) RETURN
```

```
                 IF((P[illegible].[illegible].[illegible]).[illegible].(PCL2.LT.[illegible])) PRTOUT=2
                 RETURN
          100    FORMAT(1H1,53X,'[illegible] REPORT')
          101    FORMAT(' ',54X,4('  BRIGHT  GREEN'))
          102    FORMAT(' ',9X,'[illegible] LINE PIXEL  TYPE  LABEL  CLUSTER')
          103    FORMAT(1H+,49X,'CLASS',4('  [illegible]NESS   NO. '))
          104    FORMAT(' ',8X,[illegible],[illegible],2(4X,A1),1X,2(6X,A1,A1),
               1 1X,4([illegible],3A1,4X,I4))
          105    FORMAT(' ',1[illegible]X)
          106    FORMAT('0',56X,4('ACQUISITION',I2,2X))
          200    FORMAT(I2)
          300    FORMAT(I3)
          400    FORMAT(I4)
                 END
```

FORTRAN IV[illegible] V02-5[illegible] [illegible]21149 [illegible]-AV-79 PAGE 4
DRTRAY.FTN /TR[illegible]C-E/[illegible]

PROGRAM SECTIONS

NUMBER	NAME	SIZE		ATTRIBUTES
1	ICPDE1	002574?	702	RW,I,CON,LCL
3	[illegible]A	000?4?	1??	RW,D,CON,LCL
4	[illegible]	[illegible]	77	RW,D,CON,LCL
5	[illegible]RS	000??2	1	RW,D,CON,LCL
6	[illegible]	??5?4?	?573	RW,D,OVR,GBL
7	[illegible]Y	000??4	2	RW,D,OVR,GBL
8	[illegible]	?0?A??	2?9	RW,D,OVR,GBL
9	P[illegible]T	00000?	2	RW,D,OVR,GBL

ENTRY POINTS

NAME	TYPE	ADDRESS	NAME	TYPE	ADDRESS	NAME	TYPE	ADDRESS	NAME	TYPE	ADDRESS	NAME	TYPE	ADDRESS
D[illegible]AY		1-0[illegible]												

VARIABLES

NAME	TYPE	ADDRESS	NAME	TYPE	ADDRESS	NAME	TYPE	ADDRESS	NAME	TYPE	ADDRESS	NAME	TYPE	ADDRESS
APPP	I*?	6-016[illegible]	A[illegible]	I*2	6-01[illegible]	BACATS	I*2	6-016032	CATAT	I*2	6-023352	BLK	I*1	4-000136
BUCATT	I*?	6-0142[illegible]	[illegible]	I*?	[illegible]	CHARUP	I*?	7-000000	CLUMUM	I*2	7-000002	D[illegible]	I*	6-014200
DU	I*?	6-0142??	FILE	I*2	F-[illegible]*	GRN*	I*?	4-000206	GRN2	I*2	4-0002?0	GRN3	I*2	4-000212
GRN4	I*?	4-0002??	I	I*2	[illegible]	I?	I*?	4-0002?4	K	I*2	4-000144	L	I*2	4-000216
LB	L*?	4-000[illegible]	L[illegible]	I*2	4-00?17?	LFL[illegible]	I*2	4-00?1A2	LI??	I*2	0-000000	N	I*2	?-000220
N	I*?	[illegible]	[illegible]	I*2	[illegible]	[illegible]	I*2	4-000222	[illegible]	L*1	4-000?37	[illegible]	L*1	4-000140
PACE	I*?	2-000??2	PASS	I*2	4-0??134	PCC31	R*4	6-0161??	PCC32	R*4	6-016104	PCC4	R*4	6-016070
PCC?	R*?	6-01????	PCL?1	R*4	?-01?120	PCLP2	R*4	6-016124	PCL?	R*4	6-016110	PCL2	R*4	6-016114
PRT?UT	I*?	F-00[illegible]	C	I*2	4-0??224	R	I*2	F-000024*	REC	I*2	4-000040	RJ	I*2	?-000042
SETSR	I*?	4-00[illegible]	BLANK	L*1	4-000141	SLG1	I*2	4-000172	SLG2	I*2	4-000174	SLG3	I*2	4-000176
SLG4	I*?	4-0002??	[illegible]	I*2	4-00[illegible]	TO	I*2	6-014214	TH1	I*2	4-000156	TH2	I*2	?-000160
TH?	L*?	4-00[illegible]	U	I*	4-00???									

ARRAYS

NAME	TYPE	ADDRESS	SIZE		DIMENSIONS
A	R*4	?-01??2?	[illegible]	12	(3,2)
BACAIT	R*4	6-00?0??	??2??	13?2	(2?,2?)
BACLAP	L*1	4-01?[illegible]	[illegible]	13	(2?)
BA[illegible]	I*?	6-0[illegible]	??0?74	??	(3?)
BALAR?	L*1	4-014?7?	?0??72	13	(2?)
BA[illegible]	I*?	[illegible]	[illegible]	[illegible]	([illegible])
BARA?S	R*4	?-01[illegible]	??1??	6?	(3?)
BARC??	R*4	6-0?2?7?	?24?0?	2?6	(2,8,?)
BATC??	R*4	6-02[illegible]	?0?? ?	32	(2,?)
BAL?C?	I*?	?-0?0?7?	?00?74	3?	(3?)
BAVAR	R*4	?-0??2??	?0?172	6?	(3?)
BCV?	I*?	?-01??72	[illegible]	?3?	(2?,2?)
BCV?	I*?	?-02?4?2	?0??	676	(2?,2?)

[illegible] [illegible] FC-S[illegible]

[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	[illegible]
[illegible]	[illegible]	[illegible]	[illegible]	13	(2[illegible])
[illegible]	[illegible]	6=[illegible]	[illegible]074	[illegible]	(3[illegible])
[illegible]	[illegible]	6=[illegible]4242	[illegible]	13	([illegible])
BCPTP	[illegible]	[illegible]=[illegible]0074	[illegible]0074	[illegible]	(2[illegible])
[illegible]	[illegible]	[illegible]=[illegible]	[illegible]	[illegible]	(3[illegible])
[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	([illegible])
[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	([illegible])
[illegible]	[illegible]	[illegible]	[illegible]074	[illegible]	([illegible])
[illegible]	[illegible]	6=[illegible]	[illegible]	[illegible]	(3[illegible])
[illegible]	[illegible]	6=[illegible]	[illegible]	[illegible]	([illegible])
[illegible]	[illegible]	4=[illegible]	[illegible]	4	(2)
[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	([illegible])
[illegible]	L[illegible]	8=[illegible]	[illegible]642	[illegible]	([illegible]18)
CLUD	[illegible]	6=[illegible]	[illegible]	4	(2)
C[illegible]L	[illegible]	4=[illegible]	[illegible]	12	(15)
DT	L[illegible]	[illegible]	[illegible]	2	([illegible])
GRAINS	L[illegible]	6=[illegible]	[illegible]	13	(2[illegible])
[illegible]	[illegible]	[illegible]	[illegible]	[illegible]	(1)
LABEL	L[illegible]	4=[illegible]	[illegible]	2	(5)
L[illegible]	[illegible]	6=[illegible]	[illegible]3[illegible]	1[illegible]	(2[illegible])
L[illegible]	[illegible]	4=[illegible]	[illegible]	3	(4)
[illegible]T	[illegible]	3=[illegible]	[illegible]	5	([illegible])
RC[illegible]S	[illegible]	[illegible]	[illegible]	2[illegible]	([illegible])
S	[illegible]	[illegible]	[illegible]	12	([illegible])
TC2[illegible]5	[illegible]	6=[illegible]	[illegible]	3[illegible]	(2,[illegible])
TYPE	[illegible]	6=[illegible]	[illegible]321	104	(2[illegible]9)
VC	[illegible]	4=[illegible]	[illegible]	4	(2)

LABELS

LABEL	ADDRESS	LABEL	ADDRESS	LABEL	ADDRESS	LABEL	ADDRESS	LABEL	ADDRESS
1	1=[illegible]578	2	1=[illegible]	3	1=002366	6	1=000A64	9	1=000714
10	1=[illegible]14[illegible]	11	1=[illegible]2[illegible]	2[illegible]	1=[illegible]2[illegible]	100[illegible]	3=[illegible]	101[illegible]	3=000 22
1[illegible]	3=[illegible]	1[illegible]	3=[illegible]	104[illegible]	3=[illegible]74	105[illegible]	3=[illegible]	106[illegible]	3=000762
200[illegible]	3=[illegible]31[illegible]	300[illegible]	3=[illegible]2[illegible]	400[illegible]	**				

FUNCTIONS AND SUBROUTINES REFERENCED

[illegible]

TOTAL SPACE ALLOCATED = [illegible] [illegible]

[illegible]

SEPRPT.FTN /TR:BLOCKS/WR

```
      SUBROUTINE SEPRPT(TCHAN,T2TSEP,SDATA)
      IMPLICIT INTEGER(A-Z)
      LOGICAL*1 SDATA(1),T2TSEP(1)
      WRITE(2,200)
      IF(TCHAN.NE.0) GO TO 20
      WRITE(2,199)
199   FORMAT('0','DATA FILE EMPTY')
      RETURN
20    WRITE(2,201) (I,I=1,16)
      WRITE(2,202)
      WRITE(2,203) (T2TSEP(J),J=1,8)
      DPTR=0
      DO 50 I=1,14
      DO 47 KK=1,8
      IF(SDATA(KK+DPTR).NE.'0') GO TO 48
47    CONTINUE
      GO TO 49
48    GO TO (1,2,3,4,5,6,7,8,9,10,11,12,13,14),I
1     WRITE(2,101)
      GO TO 46
2     WRITE(2,102)
      GO TO 46
3     WRITE(2,103)
      GO TO 46
4     WRITE(2,104)
      GO TO 46
5     WRITE(2,105)
      GO TO 46
6     WRITE(2,106)
      GO TO 46
7     WRITE(2,107)
      GO TO 46
8     WRITE(2,108)
      GO TO 46
9     WRITE(2,109)
      GO TO 46
10    WRITE(2,110)
      GO TO 46
11    WRITE(2,111)
      GO TO 46
12    WRITE(2,112)
      GO TO 46
13    WRITE(2,113)
      GO TO 46
14    WRITE(2,114)
46    WRITE(2,203) (SDATA(DPTR+KK),KK=1,8)
49    DPTR=DPTR+8
50    CONTINUE
      RETURN
200   FORMAT(1H1,59X,'SEPARABILITY REPORT')
201   FORMAT('0',19X,'CHANNELS',7X,16I3,6X,'SEPARABILITY')
202   FORMAT('0',19X,'AVAILABLE',6X,<TCHAN>(2X,'X'))
203   FORMAT(1H+,88X,2A1,'.',6A1)
101   FORMAT('0',19X,'1AND',11X,4(2X,'X'))
102   FORMAT('0',46X,4(2X,'X'))
103   FORMAT('0',59X,4(2X,'X'))
```

SEPRPT.FTN /TR1BLOCKS/WR

```
0057    104     FØRMAT('0',70X,4(2X,'X'))
0058    105     FØRMAT('0',19X,'3ACØ',11X,8(2X,'X'))
0059    106     FØRMAT('0',34X,4(2X,'X'),12X,4(2X,'X'))
0060    107     FØRMAT('0',34X,4(2X,'X'),24X,4(2X,'X'))
0061    108     FØRMAT('0',46X,8(2X,'X'))
0062    109     FØRMAT('0',46X,4(2X,'X'),12X,4(2X,'X'))
0063    110     FØRMAT('0',58X,8(2X,'X'))
0064    111     FØRMAT('0',19X,'3ACØ',11X,12(2X,'X'))
0065    112     FØRMAT('0',36X,8(2X,'X'),12X,4(2X,'X'))
0066    113     FØRMAT('0',36X,4(2X,'X'),12X,8(2X,'X'))
0067    114     FØRMAT('0',46X,12(2X,'X'))
0068            END
```

SEPRPT,FTN /TR:BLOCKS/WR

PROGRAM SECTIONS

NUMBER	NAME	SIZE		ATTRIBUTES
1	$CODE1	001322	361	RW,I,CON,LCL
2	$DATA	000036	15	RW,D,CON,LCL
3	$IDATA	000640	208	RW,D,CON,LCL
4	$VARS	000010	4	RW,D,CON,LCL

ENTRY POINTS

NAME	TYPE	ADDRESS	NAME	TYPE	ADDRESS	NAME	TYPE	ADDRESS	NAME	TYPE	ADDRESS	NAME	TYPE	ADDRESS
SEPRPT		1-000000												

VARIABLES

NAME	TYPE	ADDRESS	NAME	TYPE	ADDRESS	NAME	TYPE	ADDRESS	NAME	TYPE	ADDRESS	NAME	TYPE	ADDRESS
DPTR	I*2	4-000004	I	I*2	4-000000	J	I*2	4-000002	KK	I*2	4-000006	TOTAL	I*2	3-000636

ARRAYS

NAME	TYPE	ADDRESS	SIZE		DIMENSIONS
SDATA	L*1	F-000006*	000001	0	(1)
TOTSEP	L*1	F-000004*	000001	0	(1)

LABELS

LABEL	ADDRESS	LABEL	ADDRESS	LABEL	ADDRESS	LABEL	ADDRESS	LABEL	ADDRESS
1	1-000444	2	1-000476	3	1-000537	4	1-000567	5	1-000617
6	1-000642	7	1-000672	8	1-000729		1-000753	10	1-001002
11	1-001032	12	1-001062	13	1-001117	14	1-001143	20	1-001134
46	1-001170	47	**	48	1-000422	49	1-001254	50	**
101'	3-000174	102'	3-000222	103'	3-000240	104'	3-000256	105'	3-000274
106'	3-000322	107'	3-000352	108'	3-000402	109'	3-000420	110'	3-000451
111'	3-000466	112'	3-000514	113'	3-000544	114'	3-000571	199'	3-000608
200'	3-000024	201'	3-000056	202'	3-000122	203'	3-000154		

TOTAL SPACE ALLOCATED = 002236 588

NO FPP INSTRUCTIONS GENERATED

SEPRPT,LP1=SEPRPT

```
CALC.FTN          /TR:BL0CKS/WR
      SUBROUTINE CALC
      IMPLICIT INTEGER(A-Z)
      INCLUDE 'COMMON1.DAT'
      BYTE CLASFY(418),CLDAT(418)
      LOGICAL*1 TYPE(209),LBLED(209)
      LOGICAL*1 BALAPL(8),BACLAB(8),BUCLAB(8),BULAPL(8)
      REAL PCC1,PCC2,PCCG1,PCCG2,PCL1,PCL2,PCLG1,PCLG2
      REAL CLAD,CLUD,TCONS,RCONS,CCONS
      DIMENSION CLAD(2),CLUD(2),TCONS(2,8),CCONS(2),RCONS(2,8,8)
      DIMENSION PCVA(8,8),PCVU(8,8)
      DIMENSION BAPOP(12),BUPOP(12)
      COMMON/CBIA /BAPOP,BUPOP,BAUNCO,BUUNCO,BACOR,BUCOR,BACAIN,BUCAIN,
     *            BAVAR,BUVAR,BARANS,BURANS,A,DO,DU,TC,RUCATS,BUCLAB,
     *            BULAPL,BALAPL,CLASFY,TYPE,LBLED,BACATS,BACLAB,
     *            PCC1,PCC2,PCCG1,PCCG2,PCL1,PCL2,PCLG1,PCLG2,
     *            PCVA,PCVU,BATOT,BLTOT,CLAD,CLUD,TCONS,RCONS,CCONS
      COMMON/DUMMY/CLADUM,CLUDUM
      COMMON/CLCOM/CLDAT
      REAL A(3,2)
      DIMENSION BAUNCO(12),BUUNCO(12),BACOR(12),BUCOR(12)
      REAL BACAIN(8,8),BUCAIN(8,8),BAVAR(12),BUVAR(12),BARANS(12),BURANS(12)
      DIMENSION LCAT(11),BCVT(8)
      REAL PSUM,VAR,ALM,ALG,BASE1,BASE2,V3F,BASEF
      REAL ALNTF,ALNOF,ALGTF,ALGOF,GAPF,GASF,Z4F
      REAL NOCLF,LCATF(12),SLK,CLK,KARY(8),BAPOPF(12)
      BYTE LB,CL,C(12),CL2
      NTYP2=0
      NTYP1=0
      NAII=0
      NBII=0
      NGIJ=0
      BASE=22932
      GNS=0
      BAPOP(12)=0
      BAVAR(12)=0
      BACOR(12)=0
      BAUNCO(12)=0
            BARANS(12)=0
      W=0
      S=0
      N=0
      CHECK=0
      PCC1=0
      PCC2=0
      NOCL=0
      PCCG1=0
      PCCG2=0
      IF(CLADUM.EQ.1) GO TO 475
      DO 15 I=1,BACATS
      IF(BACLAB(I).NE.'W') GO TO 10
      CHECK=CHECK+1
      W=I
      GO TO 15
   10 IF(BACLAB(I).NE.'S') GO TO 12
      CHECK=CHECK+1
      S=I
```

```
0053                GØ TØ 15
0054     12         IF(BACLAR(I),NE,'N') GØ TØ 15
0055                N=1
0056     15         CØNTINUE
0057                DØ 40 I=1,11
0058                LCAT(I)=0
0059     40         CØNTINUE
0060                DØ 401 J=1,8
0061                DØ 401 I=1,8
0062                RCVA(J,I)=0
0063     401          CØNTINUE
0064                DØ 41 I=1,12
0065                PARAMS(I)=0
0066     41         CØNTINUE
0067                DØ 39 I=1,200
0068                LB=LABELED(I)
0069                CL=CLASFY(2*I-1)
0070                CL2=CLASFY(2*I)
0071                IF(LB .EQ. ' ') GØ TØ 39
0072                IF (CL .EQ.'0' .ØR.CL.EQ.' ') GØ TØ 39
0073                IF (TYPE(I).EQ.'2'.ØR.TYPE(I).EQ.'0') GØ TØ 35
                C
                C CØMPUTE TYPE 1 DØT DATA
                C
0074                NTYP1=NTYP1+1
0075                IF (LB.NE.CL .AND. CL.NE.' ') GØ TØ 32
0076                NAII=NAII +1
0077                GØ TØ 35
0078     32         IF(CHECK.LT.2)GØ TØ 35
0079     33         IF (LB.EQ.BACLAB(N).AND.CL.EQ.BACLAB(S)) NGIJ=NGIJ+1
0080                IF(LB.EQ.BACLAB(S).AND.CL.EQ.BACLAB(N)) NGIJ=NGIJ+1
                C
                C CØMPUTE TYPE 2 DØT
                C
0081     35         IF (TYPE(I).NE.'2')GØ TØ 39
0082                IF(CL2.NE.' ') GØ TØ 39
0083                NTYP2=NTYP2+1
0084                IF(LB.NE.' '.AND.CL.NE.' ') NØCL=NØCL+1
                C
                C CØNSTRUCT BIAS CØRRECTIØN VECTØRS(LB)
                C
0085                I1=0
0086                I2=0
0087                DØ 43 K=1,NACATS
0088                IF(LB.EQ.BACLAB(K))I1=K
0089                IF(CL.EQ.BACLAB(K))I2=K
0090     43         CØNTINUE
0091                IF(I1.EQ.0.ØR.I2.EQ.0)GØ TØ 39
0092                RCVA(I1,I2)=RCVA(I1,I2)+1
0093     39         CØNTINUE
                C
                C CØMPUTE GRAIN PØPULATIØN, CLASSIFIED %, CØRRECTED %
                C
0094                IF(CHECK.LT.2)GØ TØ 25
0095                BAPØP(12)=BAPØP(N)+BAPØP(S)
0096                BAUNCØ(12)=BAUNCØ(N)+BAUNCØ(S)
```

```
                C
                C COMPUTE GRAIN CORRECTED % AND VARIANCE %
                C
0097        47      ALGT=BCVA(W,W)+BCVA(W,S)+BCVA(S,W)+BCVA(S,S)
0098                ALGB=0
0099                DØ 155 I=1,BACATS
0100                ALGB=ALGB+BCVA(I,W)+BCVA(I,S)
0101        155     CONTINUE
0102                ALGTF=FLØAT(ALGT)
0103                ALGBF=FLØAT(ALGB)
0104                ALG=ALGTF/ALGBF
0105                ALNB=0
0106                ALNT=0
0107                DØ 156 I=1,BACATS
0108                IF(I.EQ.W.ØR.I.EQ.S)GØ TØ 156
0109                DØ 156 J=1,BACATS
0110                IF(J.EQ.W.ØR.J.EQ.S)GØ TØ 156
0111                ALNT=ALNT+BCVA(I,J)
0112        156     CØNTINUE
0113                DØ 158 I=1,BACATS
0114                DØ 158 J=1,BACATS
0115                    IF(J.EQ.W.ØR.J.EQ.S)GØ TØ 158
0116                ALNB=ALNB+BCVA(I,J)
0117        158     CØNTINUE
0118                BASEF=FLØAT(BASE-BAPØP(11))
0119                BASE1=0.
0120                IF(BASEF.LE.0) BASEF=1
0121                BAPF=FLØAT(BAPØP(12))
0122                BASE1=BAPF/BASEF
0123                ALNTF=FLØAT(ALNT)
0124                ALNBF=FLØAT(ALNB)
0125                IF(ALNBF.LE.0) ALNBF=1
0126                ALN=ALNTF/ALNBF
0127                N3F=0.
0128                DØ 159 I=1,BACATS
0129                IF(I.EQ.W.ØR.I.EQ.S)GØ TØ 159
0130                N3F=N3F+BAPØP(I)
0131        159     CØNTINUE
0132                BACØP(12)=(ALG*BASE1+(1.-ALN)*N3F/BASEF)*100.
0133                BASE2=(N3F/BASEF*100.)**2
0134                Z4=0
0135                DØ 21 J=1,BACATS
0136                Z4=Z4+BCVA(J,W)+BCVA(J,S)
0137        21      CØNTINUE
0138                VAR=0.
0139                Z4F=FLØAT(Z4-1)
0140                IF (Z4.LT.1)GØ TØ 24
0141                VAR=((BASE1*100.)**2*(ALG*(1.-ALG)))/Z4F
0142        24      IF(ALNBF .GT. 1.) GØ TØ 22
0143                BAVAR(12)=VAR
0144                GØ TØ 25
0145        22      BAVAR(12)=VAR+BASE2*ALN*(1.-ALN)/(ALNBF-1.)
                C
                C CØMPUTE THE RANDØM SAMPLE FØR EACH CATEGØRY EXCEPT FØR GRAIN
                C
0146        25          BATØT=0
```

```
CALC.FTN        /TR:PLØCKS/WR
0147            PSUM=0.
0148            DØ 160 I=1,BACATS
0149            PSUM=PSUM+FLØAT(APØP(I))
0150    160         CØNTINUE
0151            PSUM=PSUM/(22932.*APØP(11))
0152            NØCLF=FLØAT(NØCL)
0153            DØ 165 I=1,BACATS
0154            BCVT(I)=0
0155            DØ 168 J=1,BACATS
0156            BCVT(I)=BCVT(I)+BCVA(I,J)
0157    168     CØNTINUE
0158            BARANS(I)=FLØAT(BCVT(I))/NØCLF*PSUM*100.
0159            BATØT=BATØT+BCVT(I)
0160            IF(I.NE.W.AND.I.NE.S)GØ TØ 165
0161            GWS=GWS+BCVT(I)
0162    165     CØNTINUE
                C
                C CØMPUTE THE RANDØM SAMPLE FØR GRAIN
                C
0163            IF(CHECK.LT.2)GØTØ 100
0164            GWSF=FLØAT(GWS)
0165            BARANS(12)=GWSF/NØCLF*PSUM*100.
                C
                C CØMPUTE PCC1, PCC2, PCCG1, PCCG2
0166    100     BATAT=0
0167            X=0
0168            IF (NTYP1 .EQ. 0) GØ TØ 45
0169            PCC1=FLØAT(NAIT)*100./FLØAT(NTYP1)
0170    45      IF(NTYP2 .EQ. 0) GØ TØ 469
0171            X=BCVA(1,1)+BCVA(2,2)+BCVA(3,3)+BCVA(4,4)+BCVA(5,5)
0172            X=X+BCVA(6,6)+BCVA(7,7)+BCVA(8,8)
0173            PCC2=FLØAT(X)*100./FLØAT(NTYP2)
0174    469     IF(CHECK.LT.2)GØ TØ 470
0175            BATAT=BCVA(W,S)+BCVA(S,W)
0176            DØ 175 I=1,BACATS
0177            BATAT=BATAT+BCVA(I,I)
0178    175     CØNTINUE
0179                IF (NTYP1.EQ.0)GØ TØ 468
0180            PCCG1=(FLØAT(NAIT)+FLØAT(NGIJ))*100./FLØAT(NTYP1)
0181    468     IF(NTYP2.EQ.0)GØ TØ 470
0182            PCCG2=FLØAT(BATAT)*100./FLØAT(NTYP2)
0183    470     CØNTINUE
                C CLADW CALCULATIØNS FIRST
0184            DØ 280 MM=1,2
0185            IF(MM.EQ.1) KK=W
0186            IF(MM.EQ.2) KK=S
0187            SLW=0.
0188            CLW=0.
0189            IF(KK.EQ.0) GØ TØ 270
0190            WARY(7)=0.
0191            WARY(1)=RACØR(KK)
0192            WARY(2)=BAUNCØ(KK)
0193            WARY(3)=BARANS(KK)
0194            WARY(4)=RAVAR(KK)
0195            WARY(5)=RACAIN(KK,KK)
0196            WARY(6)=PCC1
```

```
CALC.FTN        /TR1BLØCKS/WR
0197            IF(N.NE.0) WARY(7)=BACAIN(N,N)
0198             ARY(8)=PC**2
0199            DØ 250 I=1,8
0200            SLW=SLW+WARY(I)*TCØNS(MM,I)
0201            DØ 240 J=1,8
0202            CLW=CLW+WARY(I)*WARY(J)*RCØNS(MM,I,J)
0203    240     CØNTINUE
0204    250     CØNTINUE
0205            SLW=SLW+CCØNS(MM)
0206    270     CLAD(MM)=CLW+SLW
0207    280     CØNTINUE
0208            NTYP2=0
0209            NAII=0
0210            NBII=0
0211            NGIJ=0
0212            BASE=22932
0213            GWS=0
0214            BUP2P(12)=0
0215           BUVAR(12)=0
0216            BUC2R(12)=0
0217            BUUNCØ(12)=0
0218                 BURANS(12)=0
0219            N=0
0220            S=0
0221            CHECK=0
0222            PCL1=0
0223            NØCL=0
0224            NTYP1=0
0225            PCL2=0
0226            PCLC1=0.
0227            PCLC2=0.
0228            IF(CLUDUM.EQ.1) RETURN
0229            DØ 915 I=1,BUCATS
0230            IF(BUCLAR(I).NE.'N') GØ TØ 910
0231            CHECK=CHECK+1
0232            N=I
0233            GØ TØ 915
0234    910     IF(BUCLAR(I).NE.'S') GØ TØ 915
0235            CHECK=CHECK+1
0236            S=I
0237    915     CØNTINUE
0238            DØ 940 I=1,11
0239            LCAT(I)=0
0240    940     CØNTINUE
0241            DØ 9401 J=1,8
0242            DØ 9401 I=1,8
0243            BCVU(J,I)=0
0244    9401      CØNTINUE
0245            DØ 941 I=1,12
0246            BURANS(I)=0
0247    941     CØNTINUE
0248            DØ 939 I=1,209
0249            LB=LABLED(I)
0250            CL=CLDØT(2*I-1)
0251            CL2=CLDØT(2*I)
0252            IF(LB.EQ.' ') GØ TØ 939
```

```
CALC.FTN        /TR:BLØCKS/WD
0253            IF (CL .EQ.'0' .OR.CL.EQ.' ') GØ TØ 939
0254            IF (TYPE(I).EQ.'2'.ØR.TYPE(I).EQ.'D') GØ TØ 935
                C
                C CØMPUTE TYPE 1 DØT DATA
                C
0255            NTYP1=NTYP1+1
0256            IF (LB.NE.CL .AND. CL.NE.' ') GØ TØ 932
0257            NAII=NAII +1
0258            GØ TØ 935
0259      932   IF(CHECK.LT.2)GØ TØ 935
0260      933   IF (LB.EQ.BUCLAB(W).AND.CL.EQ.BUCLAB(S)) NGIJ=NGIJ+1
0261            IF(LB.EQ.BUCLAB(S).AND.CL.EQ.BUCLAB(W)) NGIJ=NGIJ+1
                C
                C CØMPUTE TYPE 2 DØT
                C
0262      935   IF (TYPE(I).NE.'2')GØ TØ 939
0263            IF(CL2.NE.' ') GØ TØ 939
0264            NTYP2=NTYP2+1
0265            IF(LB.NE.' '.AND.CL.NE.' ') N2CL=NØCL+1
                C
                C CØNSTRUCT BIAS CØRRECTIØN VECTØRS(LB)
                C
0266            Z1=0
0267            Z2=0
0268            DØ 943 K=1,BUCATS
0269            IF(LB.EQ.BUCLAB(K))Z1=K
0270            IF(CL.EQ.BUCLAB(K))Z2=K
0271      943   CØNTINUE
0272            IF(Z1.EQ.0.ØR.Z2.EQ.0)GØ TØ 939
0273            BCVU(Z1,Z2)=BCVU(Z1,Z2)+1
0274      939   CØNTINUE
                C
                C CØMPUTE GRAIN PØPULATIØN, CLASSIFIED %, CØRRECTED %
                C
0275            IF(CHECK.LT.2)GØ TØ 925
0276            BUPØP(12)=BUPØP(W)+BUPØP(S)
0277            BUUNCØ(12)=BUUNCØ(W)+BUUNCØ(S)
                C
                C CØMPUTE GRAIN CØRRECTED % AND VARIANCE %
                C
0278      947     ALGT=BCVU(W,W)+BCVU(W,S)+BCVU(S,W)+BCVU(S,S)
0279            ALGR=0
0280            DØ 9155 I=1,BUCATS
0281            ALGR=ALGR+BCVU(I,W)+BCVU(I,S)
0282      9155  CØNTINUE
0283            ALGTF=FLØAT(ALGT)
0284            ALGRF=FLØAT(ALGR)
0285            ALG=ALGTF/ALGRF
0286            ALNR=0
0287            ALNT=0
0288            DØ 9156 I=1,BUCATS
0289            IF(I.EQ.W.ØR.I.EQ.S)GØ TØ 9156
0290            DØ 9156 J=1,BUCATS
0291            IF(J.EQ.W.ØR.J.EQ.S)GØ TØ 9156
0292            ALNT=ALNT+BCVU(I,J)
0293      9156  CØNTINUE
```

```
CALC,FTN          /TRIBLØCKS/WP
0294              DØ 9158 I=1,NUCATS
0295              DØ 9158 J=1,NUCATS
0296                 IF(J.EQ.W.ØR.J.EQ.S)GØ TØ 9158
0297              ALNR=ALNR+RCVU(I,J)
0298    9158      CØNTINUE
0299              BASEF=FLØAT(BASE-BUPØP(11))
0300              BASE1=0.
0301              IF(BASEF.LE.0) BASEF=1
0302              BAPF=FLØAT(BUPØP(12))
0303              BASE1=BAPF/BASEF
0304              ALNTF=FLØAT(ALNT)
0305              ALNBF=FLØAT(ALNB)
0306              IF(ALNBF.LE.0) ALNBF=1
0307              ALN=ALNTF/ALNBF
0308              N3F=0.
0309              DØ 9159 I=1,NUCATS
0310              IF(I.EQ.W.ØR.I.EQ.S)GØ TØ 9159
0311                 N3F=N3F+FLØAT(BUPØP(I))
0312    9159      CØNTINUE
0313              BUCVR(12)=(ALG*(BASE1+(1.-ALN)*N3F/BASEF)*100.
0314              BASE2=(N3F/BASEF*100.)**2
0315              Z4=0
0316              DØ 921 J=1,NUCATS
0317              Z4=Z4+RCVU(J,W)+RCVU(J,S)
0318    921       CØNTINUE
0319              VAR=0.
0320              Z4F=FLØAT(Z4-1)
0321              IF (Z4.LT.1)GØ TØ 924
0322              VAR=((BASE1*100.)**2*(ALG*(1.-ALG)))/Z4F
0323    924       IF(ALNBF .GT. 1.) GØ TØ 922
0324              BUVAR(12)=VAR
0325              GØ TØ 925
0326    922        BUVAR(12)=VAR+BASE2*ALN*(1.-ALN)/(ALNBF-1.)
                C
                C CØMPUTE THE RANDØM SAMPLE FØR EACH CATEGØRY EXCEPT FØR GRAIN
                C
0327    925                BUTØT=0
0328              PSUM=0.
0329              DØ 9160 I=1,NUCATS
0330              PSUM=PSUM+FLØAT(BUPØP(I))
0331    9160         CØNTINUE
0332              PSUM=PSUM/(22932.-BUPØP(11))
0333              NCLF=FLØAT(NCL)
0334              DØ 9165 I=1,NUCATS
0335              RCVT(I)=0
0336              DØ 9168 J=1,NUCATS
0337              RCVT(I)=RCVT(I)+RCVU(I,J)
0338    9168      CØNTINUE
0339              BURANS(I)=FLØAT(RCVT(I))/NCLF*PSUM*100.
0340              BUTØT=BUTØT+RCVT(I)
0341              IF(I.NE.W.AND.I.NE.S)GØ TØ 9165
0342              GWS=GWS+RCVT(I)
0343    9165      CØNTINUE
                C
                C CØMPUTE THE RANDØM SAMPLE FØR GRAIN
                C
```

```
CALC,FTN        /TR:BLOCKS/WR
0344            IF(CHECK.LT.2)GO TO 9100
0345            GWSF=FLOAT(GWS)
0346            RUNAWS(12)=GWSF/AOCLF*PSUM*100.
                C
                C COMPUTE PCC1, PCC2, PCCG1, PCCG2
0347    9100    BATAT=0
0348            X=0
0349            IF (NTYP1 .EQ. 0) GO TO 945
0350            PCL1=FLOAT(NAIT)*100./FLOAT(NTYP1)
0351    945     IF(NTYP2 .EQ. 0) GO TO 9469
0352            X=RCVU(1,1)+RCVU(2,2)+RCVU(3,3)+RCVU(4,4)+RCVU(5,5)
0353            X=X+RCVU(6,6)+RCVU(7,7)+RCVU(8,8)
0354            PCL2=FLOAT(X)*100./FLOAT(NTYP2)
0355    9469     IF(CHECK.LT.2)GO TO 9470
0356            BATAT=RCVU(W,S)+RCVU(S,W)
0357            DO 9175 I=1,RUNAWS
0358            BATAT=BATAT+RCVU(I,I)
0359    9175    CONTINUE
0360                IF (NTYP1.EQ.0)GO TO 9468
0361            PCLG1=(FLOAT(NAIT)+FLOAT(NGIJ))*100./FLOAT(NTYP1)
0362    9468     IF(NTYP2.EQ.0)GO TO 9470
0363            PCLG2=FLOAT(BATAT)*100./FLOAT(NTYP2)
0364    9470    CONTINUE
                C CLOUD CALCULATIONS NOW
0365            DO 9260 MM=1,2
0366            IF(MM.EQ.1) KK=W
0367            IF(MM.EQ.2) KK=S
0368            SLW=0.
0369            CLW=0.
0370            IF(KK.EQ.0) GO TO 9270
0371            WARY(7)=0.
0372            WARY(1)=RUCOP(KK)
0373            WARY(2)=RUINC2(KK)
0374            WARY(3)=RUNAWS(KK)
0375            WARY(4)=RUVAR(KK)
0376            WARY(5)=RUCAIN(KK,KK)
0377            WARY(6)=PCL1
0378            IF(N.NE.0) WARY(7)=BUCAIN(N,N)
0379            WARY(8)=PCL2
0380            DO 9250 I=1,8
0381            SLW=SLW+WARY(I)*TCONS(MM,I)
0382            DO 9240 J=1,8
0383            CLW=CLW+WARY(I)*WARY(J)*FCONS(MM,I,J)
0384    9240    CONTINUE
0385    9250    CONTINUE
0386            SLW=SLW+CCONS(MM)
0387    9270    CLUD(MM)=CLW+SLW
0388    9280    CONTINUE
0389            CONTINUE
0390            RETURN
0391            END
```

PROGRAM SECTIONS

NUMBER	NAME	SIZE		ATTRIBUTES
1	$CODE1	011150	2376	RW,I,CON,LCL
2	$PDATA	000004	2	RW,D,CON,LCL
4	$VARS	000262	153	RW,D,CON,LCL
5	$TEMPS	000334	14	RW,D,CON,LCL
6	OTHER	000122	1321	RW,D,OVR,GBL
7	DUMMY	000004	2	RW,D,OVR,GBL
8	CLOCK	000642	209	RW,D,OVR,GBL

ENTRY POINTS

NAME	TYPE	ADDRESS	NAME	TYPE	ADDRESS	NAME	TYPE	ADDRESS	NAME	TYPE	ADDRESS	NAME	TYPE	ADDRESS
CALC		1-000000												

VARIABLES

NAME	TYPE	ADDRESS	NAME	TYPE	ADDRESS	NAME	TYPE	ADDRESS	NAME	TYPE	ADDRESS	NAME	TYPE	ADDRESS
ALG	R*4	4-000060	ALOG	I*2	4-000442	ALGHF	R*4	4-000122	ALOG	I*2	4-000442	[illegible]	I*2	4-000116
ALN	R*4	4-000056	ALOG	I*2	4-000442	ALNHF	R*4	4-000112	ALOG	I*2	4-000442	[illegible]	I*2	4-000176
BACATS	I*2	6-003314	BASE	R*4	4-000126	BASE	I*2	4-000041	BASEF	R*4	4-000112	[illegible]	I*2	4-000346
BASE2	R*4	4-000072	BATAT	I*2	4-000432	BATZT	I*2	6-003766	BACATS	I*2	6-003314	[illegible]	I*2	6-003772
CHECK	I*2	4-000422	CL	L*1	4-000357	CLARGM	I*2	7-000000	CLOCK	I*2	7-000002	[illegible]	I*2	4-000232
CL2	L*1	4-000374	DF	I*2	4-001550	DJ	I*2	6-001552	[illegible]	I*2	4-000402	[illegible]	I*2	4-000132
I	I*2	4-000426	J	I*2	4-000430	K	I*2	4-000434	KK	I*2	4-000440	[illegible]	I*1	4-000356
MM	I*2	4-000456	N	I*2	4-000420	NAII	I*2	6-000402	NPTT	I*2	4-000404	[illegible]	I*2	6-000406
NOOL	I*2	4-000424	NOOLF	R*4	4-000142	NTYP1	I*2	4-000400	NTYP2	I*2	4-000402	[illegible]	R*4	4-000076
POCO1	R*4	6-003336	POCO2	R*4	6-003342	POC1	R*4	4-000324	POCO	R*4	4-000332	[illegible]	R*4	4-000356
POLO2	R*4	6-003362	POL1	R*4	6-003346	POL2	R*4	6-003352	POLO	R*4	4-000340	[illegible]	I*2	4-000416
SL	R*4	4-000226	TT	I*2	6-001554	VAR	R*4	4-000032	[illegible]	I*2	4-000444	[illegible]	I*2	6-000454
Z1	I*2	4-000432	Z2	I*2	4-000434	Z4	I*2	4-000450	Z4F	I*2	4-000446			

ARRAYS

NAME	TYPE	ADDRESS	SIZE		DIMENSIONS
A	R*4	6-001520	000030	12	(3,2)
BACATS	R*4	6-003220	000400	128	(8,8)
BACLAB	L*1	6-003316	000010	4	(8)
BACOR	I*2	6-000140	000030	12	(12)
BALABL	L*1	6-001600	000010	4	(8)
BAPRP	I*2	6-000000	000030	12	(12)
BAPRPF	R*4	4-000276	000060	24	(12)
BARAYS	R*4	6-001350	000060	24	(12)
BAUNOT	I*2	6-000060	000030	12	(12)
BAVAR	R*4	6-001220	000060	24	(12)
BCVA	I*2	6-003366	000200	64	(8,8)
BCVT	I*2	4-000024	000020	8	(8)
BCVL	I*2	6-003566	000200	64	(8,8)

BUCAIN	R*4	6-000620	000400	128	(8,8)
BUCLAB	L*1	6-001560	000010	4	(8)
BUC2R	I*2	6-000170	000030	12	(12)
BULABL	L*1	6-001570	000010	4	(8)
BUP2P	I*2	6-000030	000030	12	(12)
BURANS	R*4	6-001440	000060	24	(12)
BUUNCO	I*2	6-000110	000030	12	(12)
BUVAR	R*4	6-001310	000060	24	(12)
C	L*1	4-000350	000014	6	(12)
CCONS	R*4	6-005112	000010	4	(2)
CLAD	R*4	6-003772	000010	4	(2)
CLASFY	L*1	6-001610	001642	209	(41B)
CLDAT	L*1	6-000000	001642	209	(41B)
CLUD	R*4	6-004002	000010	4	(2)
LBLED	L*1	6-002773	000321	104	(209)
LCAT	I*2	4-000000	000026	11	(11)
LCATF	R*4	4-000146	000060	24	(12)
RCONS	R*4	6-004112	001000	256	(2,8,8)
TCONS	R*4	6-004012	000100	32	(2,8)
TYPE	L*1	6-002452	000321	104	(209)
VARY	R*4	4-000236	000040	16	(8)

LABELS

LABEL	ADDRESS	LABEL	ADDRESS	LABEL	ADDRESS	LABEL	ADDRESS	LABEL	ADDRESS
10	1-000236	12	1-000272	15	1-000320	21	**	22	1-002450
24	1-002612	25	1-002724	32	1-000702	33	**	35	1-001000
39	1-001210	40	**	41	**	47	**	48	1-003364
47	**	100	1-003312	155	**	156	1-001460	156	1-002034
159	1-002302	160	**	165	1-003224	168	**	175	**
240	**	250	**	270	1-004440	280	**	471	**
468	1-003570	469	1-003674	470	1-003732	910	1-004720	915	1-004752
921	**	922	1-007302	924	1-007244	925	1-007356	932	1-005334
933	**	935	1-005432	939	1-005642	940	**	941	**
943	**	945	1-010016	947	**	9101	1-007744	9105	**
9156	1-006320	9158	1-006466	9159	1-006734	9160	**	9165	1-007654
9168	**	9175	**	9240	**	9250	**	9371	1-010272
9280	**	9401	**	9468	1-010322	9469	1-010424	9470	1-010364

TOTAL SPACE ALLOCATED = 017662 4057

CALC,LP:=CALC

```
      SUBROUTINE BIAPRT(IBUF,TAPN2,PASS)
      IMPLICIT INTEGER (A-Z)
      LOGICAL*1 TAPE(1)
      INCLUDE 'COMMON.DAT'
      BYTE CLASFY(418),CLRST(418)
      LOGICAL*1 TYPE(209),LBLEN(209)
      LOGICAL*1 BALABL(8),BACLAB(8),BLMLAB(8),BULABL(8)
      REAL PCC1,PCC2,PCCG1,PCCG2,PCL1,PCL2,PCLG1,PCLG2
      REAL CLAD,CLUD,ICONS,RCONS,CCONS
      DIMENSION CLAD(2),CLUD(2),ICONS(2,8),CCONS(2),RCONS(2,8,8)
      DIMENSION ICLASS(2,8),CONS(8,8)
      DIMENSION APUR(12),BUPUR(12)
      COMMON /CBIAS/BAPUR,BUPUR,BALACR,BUBACR,BACTR,BUCTR,BICAIN,BUCAIN,
     *       [illegible]
     *       BULABL,BALABL,CLASFY,TYPE,LBLEN,[illegible]
     *       PCC1,PCC2,PCCG1,PCCG2,PCL1,PCL2,PCLG1,PCLG2,
     *       [illegible],BATOT,BUTOT,CLAD,CLUD,ICONS,RCONS,CCONS
      COMMON/DUMMY/CLAD,CLUD
      COMMON/CLRST/CLRST
      REAL A(8,2)
      DIMENSION BAPURC(12),BUPURC(12),BAPUR(12),BUPUR(12)
      REAL [illegible](8,8),[illegible](8,8),BAVAR(12),BUVAR(12),[illegible](12),[illegible](12)
      COMMON/SEG/SEG
      LOGICAL*1 SEGNO(4),PI(120),BLANK,PMATR(120)
      DIMENSION COL(20)
      DATA COL/10,20,30,40,50,60,70,80,90,100,110,120,130,140,
     *   150,160,170,180,190/
      DATA BLANK/' '/,ONE/'1'/,SLASH/'/'/,TWO/'2'/
  100 FORMAT(1H1,45X,'BIAS CORRECTION CLASSIFICATION REPORT')
  101 FORMAT(1H0,20X,12(5X,'CAT'))
  102 FORMAT(1H ,19X,8(7X,1A1))
  103 FORMAT(1H0,5X,'P [illegible]    ',12(3X,I5))
  104 FORMAT(1H ,5X,'CLASSIFIED   % ',12(2X,I6))
  105 FORMAT(1H ,5X,'CORRECTED    % ',8(2X,I6),26X,I6)
  106 FORMAT(1H ,5X,'VARIANCE     % ',8(2X,F6.2),25X,F6.2)
  107 FORMAT(1H ,5X,'RANDOM SAMPLE  ',8(2X,F6.2),20X,F6.2)
  108 FORMAT(1H0,51X,'ALPHA VALUE MATRIX')
  109 FORMAT(1H ,15X,8(5X,'CAT'))
  110 FORMAT(1H ,5X,'CAT. OF INT./',5(4X,1A1,1X))
  111 FORMAT(1H ,5X,'CAT. OF INT.')
  112 FORMAT(1H ,11X,1A1,5X,8(4X,F4.3))
  113 FORMAT(1H0,35X,'BIAS CORRECTION VECTORS (LABLED:CLASSIFIED)')
  114 FORMAT(1H0,14X,8(4X,1A1,':',1A1,I3))
  115 FORMAT('1',49X,'CAMS INTERFACE REPORT')
  116 FORMAT(1H0,3X,'LOAD NO. = ',24A1,57X,'TAPE NUMBER: ',I5,'0')
  117 FORMAT(1H0,3X,'RECORD ID = ',6A1,1X,6A1)
  118 FORMAT(1H0,50X,'ACQUISITION DATE')
  119 FORMAT(1H ,47X,'1',6X,'2',6X,'3',6X,'4')
  120 FORMAT(1H ,4X,'******  SEGMENT NUMBER =',4A1,13X,4(5I1,2X))
  121 FORMAT(1H ,'TRAINING SEGMENT NO. ',I1,' =',66A1)
  122 FORMAT(1H0,42X,'TYPE',1A1,' DOT LABEL/CLASSIFICATION')
  123 FORMAT(1H0,4 X,'TYPE',1A1,' DOT LABEL/CLUSTER ASSIGNMENT')
  124 FORMAT(1H0,1X,'PIXEL/')
  125 FORMAT(' ',1X,'LINE',2X,19(I3,3X))
  126 FORMAT(1H+,97X,'DOT',4X,'TP',6X,'[illegible]',4X,'GRAI ')
  127 FORMAT(1H+,97X,'NO',5X,'CAT',6X,'UN',4X,'GRAIN')
```

```
 128  FØRMAT(1H1,41X,'BIAS CØRRECTIØN CLUSTER REPØRT')
 129  FØRMAT(1H1,13X)
 130  FØRMAT(1H+,90X,'PCC1  = ',F6.1,' PCC2  = ',F6.1)
 131  FØRMAT(1H+,90X,'PCCG1 = ',F6.1,' PCCG2 = ',F6.1)
 136  FØRMAT(1H+,90X,' DW    = ',F6.1,'  DS    = ',F6.1)
 132  FØRMAT(1H0,13X)
 133  FØRMAT(1H+,70X,'CLUSTERED ')
 134  FØRMAT(' ',20X,8(A1,'!',I1,I4,4X))
 135  FØRMAT(1H0,22X,'TØTAL = ',I5)
      GØ TØ(1,2),PASS
 1    WRITE(6,115)
      DØ 3 I=1,4
 3    SECØND(I)= BUF(147+I)
      WRITE(6,116) (IBUF(I),I=3,26),TSØNR
      WRITE(6,117) (IBUF(I),I=709,714),(IBUF(J),J=715,720)
      WRITE(6,118)
      WRITE(6,119)
      WRITE(6,120) (SECØND(I),I=1,4),(IBUF(J),J=154,158),
     2(IBUF(K),K=159,163),(IBUF(L),L=164,168),(IBUF(M),M=169,173)
      RETURN
 2    IF(CLAØUT.EQ.1) GØ TØ 2.2
      WRITE(6,129)
      CNT=ØNE
      DØ 97 JJ=1,2
      K=1
      NLINE=10
      WRITE(6,122) CNT
      WRITE(6,124)
      WRITE(6,125) (CØL(I),I=1,15)
      DØ 98 J=1,11
      L=6
      DØ 10 I=1,120
      PMATR(I)=BLANK
 10   CØNTINUE
      DØ 12 N=1,19
      IF(CNT.EQ.'1') GØ TØ 6
      IF(TYPE(K).EQ.'1'.ØR.TYPE(K).EQ.'2') GØ TØ 7
      GØ TØ 5
 6    IF(TYPE(K).EQ.'1'.ØR.TYPE(K).EQ.'3') GØ TØ 7
      GØ TØ 5
 7    PMATR(L)=LBLED(K)
      PMATR(L+1)=SLASH
      PMATR(L+2)=CLASFY(2*K)
      IF(CLASFY(2*K-1).EQ.' ') GØ TØ 5
      PMATR(L+2)=CLASFY(2*K-1)
      PMATR(L+3)=CLASFY(2*K)
 5    CØNTINUE
      K=K+1
      L=L+6
 12   CØNTINUE
      WRITE(6,99) NLINE,(PMATR(I),I=6,120)
 99   FØRMAT(1H0,2X,I3,2X,115A1)
      NLINE=NLINE+10
 98   CØNTINUE
      CNT=TWØ
 97   CØNTINUE
```

```
              WRITE(6,100)
              WRITE(6,101)
              WRITE(6,102) (PARAHL(I),I=1,6)
             WRITE(6,126)
              WRITE(6,103) (PARPP(I),I=1,12)
              WRITE(6,104) (PANCA(I),I=1,12)
              WRITE(6,105) (PACPR(I),I=1,6),BACPR(12)
              WRITE(6,106) (PAVAR(I),I=1,6),PAVAR(12)
              WRITE(6,107) (PARANS(I),I=1,3),BARANS(12)
             WRITE(6,132)
              CALL BNT
              WRITE(6,109)
             WRITE(6,130) PCC1,PCC2
              WRITE(6,109)
             WRITE(6,131) PCCS1,PCCS2
              WRITE(6,110) (PACLAB(I),I=1,8)
              WRITE(6,111)
              WRITE(6,136) (CLOD(I),I=1,2)
              DO 96 J=1,BACATS
              WRITE(6,112) BALABL(J),(BACAIN(J,K),K=1,BACATS)
      96      CONTINUE
             WRITE(6,132)
              CALL BNT
              CALL BNT
              WRITE(6,113)
             DO 50 K=1,BACATS
             WRITE(6,134) (BACLAB(K),BACLAB(LL),BCVA(K,LL),LL=1,BACATS)
      50     CONTINUE
             WRITE(6,135) BAT.Y
     202        IF(CLUDUM.EQ.1) RETURN
             WRITE(6,129)
             CNT=ONE
             DO 297 JJ=1,2
             K=1
             NLINE=10
             WRITE(6,123) CNT
             WRITE(6,124)
             WRITE(6,125) (COL(I),I=1,19)
             DO 298 J=1,11
             L=6
             DO 210 I=1,120
             PMATR(I)=BLANK
     210      CONTINUE
             DO 212 N=1,19
             IF(CNT.EQ.'1') GO TO 206
             IF(TYPE(K).EQ.'0'.OR.TYPE(K).EQ.'2') GO TO 207
             GO TO 205
     206        IF(TYPE(K).EQ.'1'.OR.TYPE(K).EQ.'3') GO TO 207
             GO TO 205
     207      PMATR(L)=LBLED(K)
             PMATR(L+1)=SLASH
             PMATR(L+2)=CLDOT(2*K)
             IF(CLDOT(2*K-1).EQ.' ') GO TO 5
              PMATR(L+2)=CLDOT(2*K-1)
              PMATR(L+3)=CLDOT(2*K)
     205      CONTINUE
```

```
0163                K=K+1
0164                L=L+6
0165     212        CONTINUE
0166                WRITE(6,99) NLINE,(PMATR(I),I=6,129)
0167                NLINE=NLINE+10
0168       299      CONTINUE
0169                CNT=T+8
0170     297        CONTINUE
0171                WRITE(6,128)
0172                WRITE(6,101)
0173                WRITE(6,102) (RUCAPL(I),I=1,8)
0174                WRITE(6,127)
0175                WRITE(6,103) (RUPPR(I),I=1,12)
0176                WRITE(6,104) (RUNOPR(I),I=1,12)
0177                WRITE(6,105) (RUCPR(I),I=1,8),RUCPR(12)
0178                WRITE(6,106) (RUVAR(I),I=1,8),RUVAR(12)
0179                WRITE(6,107) (RUPRPS(I),I=1,8),RURANS(12)
0180                WRITE(6,172)
0181                CALL BNT
0182                WRITE(6,108)
0183                WRITE(6,130) PCL1,PCL2
0184                WRITE(6,109)
0185                WRITE(6,131) PCLG1,PCLG2
0186                WRITE(6,110) (RUCLAR(I),I=1,8)
0187                WRITE(6,111)
0188                WRITE(6,136) (FLAG(I),I=1,2)
0189                DO 2? J=1,RUCATS
0190                WRITE (6,112) RULARL(J),(EUCAIN(J,K),K=1,BUCATS)
0191       2        CONTINUE
0192                WRITE(6,172)
0193                CALL BNT
0194                CALL BNT
0195                WRITE(6,113)
0196                DO 250 K=1,RUCATS
0197                WRITE(6,134) (BUCLAR(K),BUCLA3(LL),RCVU(K,LL),LL=1,BACATS)
0198     250        CONTINUE
0199                WRITE(6,135) RUTOT
0200                RETURN
0201                END
```

PROGRAM SECTIONS

NUMBER	NAME	SIZE		ATTRIBUTES
1	$CODE1	006000	1536	RW,I,CON,LCL
2	$PDATA	000006	3	RW,D,CON,LCL
3	$IDATA	001714	486	RW,D,CON,LCL
4	$VARS	000462	153	RW,D,LCN,LCL
6	CBIAS	005122	1321	RW,D,OVR,GBL
7	DUMMY	000004	2	RW,D,OVR,GBL
8	CLOCK	000642	209	RW,D,OVR,GBL
9	SEG	000004	2	RW,D,OVR,GBL

ENTRY POINTS

NAME	TYPE	ADDRESS	NAME	TYPE	ADDRESS	NAME	TYPE	ADDRESS	NAME	TYPE	ADDRESS	NAME	TYPE	ADDRESS
BIAPRT		1-000000												

VARIABLES

NAME	TYPE	ADDRESS	NAME	TYPE	ADDRESS	NAME	TYPE	ADDRESS	NAME	TYPE	ADDRESS	NAME	TYPE	ADDRESS
BACATS	I*2	6-003314	BATST	I*2	6-003766	BLANK	L*1	4-000170	BUCATS	I*2	6-001556	BUTST	I*2	6-003770
CLADUM	I*2	7-000000	CLODUM	I*2	7-000002	CNT	I*2	4-000450	DN	I*2	6-001560	I	I*2	4-001562
I	I*2	4-000436	J	I*2	4-000440	JJ	I*2	4-000452	K	I*2	4-000442	L	I*2	4-000444
LL	I*2	4-000460	M	I*2	4-000446	N	I*2	4-000456	NLINE	I*2	4-000454	Z F	I*2	4-000430
PASS	I*2	F-000006*	PCCG1	R*4	6-003336	PCCG2	R*4	6-003342	PCG1	R*4	6-003326	PCG2	R*4	6-003332
PCLG1	R*4	6-003356	PCLG2	R*4	6-003362	PCL1	R*4	6-003346	PCL2	R*4	6-003352	SLASH	I*2	4-000432
TAPN0	I*2	F-000004*	C	I*2	6-001554	TN3	I*2	4-000434						

ARRAYS

NAME	TYPE	ADDRESS	SIZE		DIMENSIONS
A	R*4	6-001520	000030	12	(3,2)
BACAIN	R*4	6-000220	000400	128	(8,8)
BACLAB	L*1	6-003316	000010	4	(8)
BAC2R	I*2	6-000140	000030	12	(12)
BALABL	L*1	6-001600	000010	4	(8)
BAP2P	I*2	6-000000	000030	12	(12)
BARANS	R*4	6-001360	000060	24	(12)
BAUNC2	I*2	6-000060	000030	12	(12)
BAVAR	R*4	6-001220	000060	24	(12)
BCVA	I*2	6-003366	000200	64	(8,8)
BCVU	I*2	6-003566	000200	64	(8,8)
BUCAIN	R*4	6-000620	000400	128	(8,8)
BUCLAB	L*1	6-001560	000010	4	(8)
BUC2R	I*2	6-000170	000030	12	(12)
BULABL	L*1	6-001570	000010	4	(8)
BUP2P	I*2	6-000030	000030	12	(12)
BURANS	R*4	6-001440	000060	24	(12)
BUUNC2	I*2	6-000110	000030	12	(12)

BIAPRT.FTN /TR:BLOCKS/WR

BVAR	R*4	6-001330	000060	24	(12)
COORS	R*4	6-005112	000010	4	(2)
CLAD	R*4	6-003772	000010	4	(2)
CLASFY	L*1	6-001610	000642	209	(416)
CLOUT	L*1	6-000000	000642	209	(416)
CLUD	R*4	6-004052	000010	4	(2)
COL	I*2	4-000352	000046	19	(19)
IBUF	L*1	5-000[illegible]	000001	1	(1)
LABLED	L*1	6-002773	000321	104	(209)
PB	L*1	4-000000	000170	60	(120)
PVATR	L*1	4-000171	000170	60	(120)
ROWS	R*4	6-004112	000000	256	(2,8,8)
SEGNO	L*1	9-000000	000004	2	(4)
TOWS	R*4	6-004012	000100	32	(2,8)
TYPE	L*1	6-002452	000321	104	(209)

LABELS

LABEL	ADDRESS	LABEL	ADDRESS	LABEL	ADDRESS	LABEL	ADDRESS	LABEL	ADDRESS
1	1-000054	2	1-000430	3	**	5	1-001764	*	1-001214
7	1-001254	10	**	12	**	20	**	21	**
96	**	97	**	98	**	99	3-[illegible]	[illegible]	[illegible]
101'	3-000054	102'	3-000074	103'	3-000112	104'	3-000150	[illegible]	[illegible]
106'	3-000230	107'	3-000314	108'	3-000360	109'	3-[illegible]	[illegible]	[illegible]
111'	3-000470	112'	3-000514	113'	3-000564	114'	**	115'	3-[illegible]
116'	3-000654	117'	3-000730	118'	3-000762	119'	3-001012	[illegible]	[illegible]
121'	**	122'	3-001114	123'	3-001164	124'	3-[illegible]	[illegible]	[illegible]
126'	3-001300	127'	3-001336	128'	3-001374	129'	3-001442	[illegible]	[illegible]
131'	3-001510	132'	3-001610	133'	**	134'	3-001614	135'	3-001642
136'	3-001550	202	1-003310	205	1-004030	206	1-[illegible]	207	1-003734
210	**	212	**	250	**	297	**	298	**

FUNCTIONS AND SUBROUTINES REFERENCED

BNT

TOTAL SPACE ALLOCATED = 016400 3712

NO FPP INSTRUCTIONS GENERATED

BIAPRT,LP:=BIAPRT

```
*FORTRAN IV-PLUS V02-51              08105155    09-MAR-78          PAGE 1
CORED.FTN       /TR:BLOCKS/WR
0001            SUBROUTINE COPED(IBUF,R,FILE)
0002            IMPLICIT INTEGER(A-Z)
0003            INTEGER*2 ISTAT(2),IPRM(6)
0004             EQUIVALENCE(ISTAT(1),IB)
0005             BYTE IB(2)
0006            LOGICAL*1 IBUF(1)
0007            DIMENSION IHDAT(1)
0008            DIMENSION IA(37)
0009            DATA XDEV/2HXT/
0010            DATA MDEV/2HMT/
0011            RECORD=R
0012            IF(RECORD.GT.0)  GO TO 10
0013             ERCNT=0
0014             IF(RECORD.EQ.-1)  GO TO 6
0015    12      CONTINUE
0016            FILE=C
        D       WRITE(5,49)                                          IOERR**
        CTYPE 49
0017    49      FORMAT(10X,' TYPE M OR X FOR TAPE DEVICE CODE'/)
0018           READ(1,51,END=99)  IA
        C       ACCEPT 51,IA
0019            IF(IA(1).EQ.'M')  GO TO 13
0020            IF(IA(1).EQ.'X')  GO TO 14
0021    99      CONTINUE
0022            WRITE(6,97)
0023    97      FORMAT(10X,' CARD ERROR  '//)
0024            WRITE(6,97)
0025            STOP
0026    13      IDEV=MDEV
0027            GO TO 15
0028    14      IDEV=XDEV
0029    15      CONTINUE
        D       WRITE(5,50)                                          IOERR**
        C       TYPE 50
0030    50      FORMAT(10X,' TYPE TAPE UNIT NUMBER = 0 OR 1')
0031            READ(1,51,END=99)  IA
        C       ACCEPT 51,IA
0032    51      FORMAT(37A2)
0033             IF(IA(1).EQ.'0')  GO TO 18
0034             IF(IA(1).EQ.'1')  GO TO 19
0035             GO TO 99
0036    18       IUNT=0
0037             GO TO 20
0038    19       IUNT=1
0039    20       CONTINUE
0040            ILUN=9
0041            IDS=0
0042            ISR=0
0043            CALL ASNLUN(ILUN,IDEV,IUNT,IDS)
0044            IF(IDS.LT.0)  GO TO 1
0045            CALL GETADR(IPRM,IBUF)
0046    6        CONTINUE
0047            CALL QIO(%2400,ILUN,1,,ISTAT,IPRM,ISR)
0048            IF(ISR.LT.0)  GO TO 2
0049             IF(RECORD.EQ.-1)  GO TO 11
0050            IPRM(2)=3600
```

```
0051     10     CONTINUE
0052            ISW=0
0053            RECORD=RECORD+1
0054            CALL QIO("1000,IUN,1,,ISTAT,IPRM,ISW)
0055            IF(ISW.LT.0)  GO TO 3
0056            CALL WAITFR(1,IDS)
0057            IF(IDS.LT.0)  GO TO 4
        C        WRITE(6,101) (IB(I),I=1,2),ISTAT(2)
0058     101    FORMAT(1H0,10X,2I4,3X,I5,2X,'BYTES TRANSFERED')
0059            ERCODE=IB(1)
0060            IF(ERCODE.LT.0)  GO TO 5
0061            FF=0
0062            DO 800 JJ=1,720
0063     798    IF(IBUF(JJ).LT.0) GO TO 799
0064     800    CONTINUE
0065            RETURN
0066     799    IBUF(JJ)='1'
0067            IF(FF.EQ.1) GO TO 798
0068            FF=1
0069            WRITE(6,1000)
0070     1000   FORMAT(1H '***** THIS SEGMENT CONTAINS BAD DATA *****'
0071            GO TO 798
0072     1      CONTINUE
0073            WRITE(6,100)  IDS
0074     100    FORMAT(1H ,'  ASLUN CALL DSW=  ',I6)
0075            STOP
0076     2      CONTINUE
0077            WRITE(6,200) ISP
0078     200    FORMAT(1H ,'REWIND DSW = ',I6)
0079            STOP
0080     3      CONTINUE
0081            WRITE(6,300)  ISW
0082     300    FORMAT(1H ,'  READ QIO DSW = ',I6)
0083            STOP
0084     4      CONTINUE
0085            WRITE(6,400)  IDS
0086     400    FORMAT(1H ,'  WAIT DSW = ',I6)
0087            STOP
0088     5      CONTINUE
0089            IF(ERCODE.EQ.-10)  FILE=FILE+1
0090            IF(ERCODE.EQ.-10)  RETURN
0091             IF(ERCODE.NE.-4)  GO TO 7
0092             ERCNT=ERCNT+1
0093             RETURN
0094     11     CONTINUE
0095            WRITE(6,700)
0096     700    FORMAT('  ')
0097            IF(ERCNT.EQ.0)  RETURN
0098            WRITE(6,600)  ERCNT
0099     600    FORMAT('1',10X,' TAPE ERRORS ENCOUNTERED = ',I5)
0100            RETURN
0101     7      CONTINUE
0102            WRITE(6,500)  ERCODE
0103     500    FORMAT(1H0,'  I/O STATUS BLOCK ERROR CODE= ',I6)
0104            STOP
0105            END
```

PROGRAM SECTIONS

NUMBER	NAME	SIZE		ATTRIBUTES
1	$CODE1	001420	392	RW,I,CON,LCL
2	$PDATA	000014	6	RW,D,CON,LCL
3	$IDATA	000472	157	RW,D,CON,LCL
4	$VARS	000164	59	RW,D,CON,LCL

ENTRY POINTS

NAME	TYPE	ADDRESS	NAME	TYPE	ADDRESS	NAME	TYPE	ADDRESS	NAME	TYPE	ADDRESS	NAME	TYPE	ADDRESS
CDRED		1-000000												

VARIABLES

NAME	TYPE	ADDRESS	NAME	TYPE	ADDRESS	NAME	TYPE	ADDRESS	NAME	TYPE	ADDRESS	NAME	TYPE	ADDRESS
ERCNT	I*2	4-000142	ERCODE	I*2	4-000160	FF	I*2	4-000162	FILE	I*2	F-000000*	IDS	I*2	4-000144
IDS	I*2	4-000152	ILUN	I*2	4-000151	ISR	I*2	4-000154	IUN	I*2	4-000165	IUNIT	I*2	4-000146
JJ	I*2	4-000164	MREV	I*2	4-000136	R	I*2	F-000004*	RECORD	I*2	4-000140	XTN	I*2	4-000134

ARRAYS

NAME	TYPE	ADDRESS	SIZE		DIMENSIONS
IA	I*2	4-000022	000112	37	(37)
IB	L*1	4-000000	000002	1	(2)
IBUF	L*1	F-000002*	000001	1	(1)
IHDAT	I*2	4-000020	000002	1	(1)
IPRM	I*2	4-000004	000014	6	(6)
ISTAT	I*2	4-000000	000004	2	(2)

LABELS

LABEL	ADDRESS	LABEL	ADDRESS	LABEL	ADDRESS	LABEL	ADDRESS	LABEL	ADDRESS
1	1-000774	2	1-001340	3	1-001104	4	1-001150	5	1-001214
6	1-000472	7	1-001354	10	1-000544	11	1-001264	12	**
13	1-000232	14	1-000250	15	1-000264	16	1-001352	17	1-000364
20	1-000402	49'	**	50'	**	51'	3-000024	97'	3-000000
99	1-000156	100'	3-000110	101'	**	200'	3-000142	300'	3-000166
400'	3-000216	500'	3-000314	600'	3-000250	700'	3-001240	798	1-000452
799	1-000712	800	**	1000'	3-000030				

FUNCTIONS AND SUBROUTINES REFERENCED

ASNLUN GETADR QIO WAITFR

```
CORED,FT5        /TR:BLOCKS/NA
TOTAL SPACE ALLOCATED = 000314   614
@
NO FPP INSTRUCTIONS GENERATED
@
CORED,LP:=CORED
```

```
COMRED.FTN      /TR:BLOCKS/WR
        SUBROUTINE COMRED
        IMPLICIT INTEGER(A-Z)
        INCLUDE 'COMMON1.DAT'
        BYTE CLASFY(415),OLDMAT(415)
        LOGICAL*1 TYPE(2,5),LBLED(255)
        LOGICAL*1 RALAPL(5),RACLAB(5),PCLAB(3),PCLAPL(5)
        REAL PCC1,PCC2,PCCG1,PCCG2,PCL1,PCL2,PCLG1,PCLG2
        REAL CLAD,CLUD,TCONS,RCONS,CCONS
        DIMENSION CLAD(2),CLUD(2),TCONS(2,8),CCONS(2),RCONS(2,8,6)
        DIMENSION RCVA(8,6),RCVU(8,6)
        DIMENSION RAPRP(12),SUPRP(12)
        COMMON/CRIAS/RAPRP,SUPRP,BALACZ,AUDACZ,BACRF,SUCRF,RACAT,RUCAT,
     1          RAVAR,SUVAR,RARANS,SURANS,A,PZ,IU,TC,SUCATS,[illegible],
     2          RUBAL,BALABL,CLASFY,TYPE,LBLED,RACATS,RACLAB,
     3          PCC1,PCC2,PCCG1,PCCG2,PCL1,PCL2,PCLG1,PCLG2,
     4          RCVA,RCVU,RATET,BLTZT,CLAD,CLUD,TCONS,RCONS,CCONS
        COMMON/DUMMY/CLADUM,CLUDUM
        COMMON/CLDAM/CLDAT
        REAL A(3,2)
        DIMENSION RANCR(12),RUNCR(12),RACRP(12),RUCRP(12)
        REAL RACBAL(8,8),RUCBAL(8,8),BALVAR(12),SUVAR(12),RARANS(12),SURANS(12)
        IU=1
        DO 10 I=1,2
        READ(IU,101) CCONS(I)
101     FORMAT(F10.5)
        READ(IU,100) (TCONS(I,J),J=1,8)
        DO 9 J=1,8
        READ(IU,100) (RCONS(I,J,K),K=1,6)
9       CONTINUE
10      CONTINUE
100     FORMAT(8F10.5)
        RETURN
        END
```

PROGRAM SECTIONS

NUMBER	NAME	SIZE		ATTRIBUTES
1	$CODE1	000334	110	RW,I,CON,LCL
3	$DATA	000010	5	RW,D,CON,LCL
4	$VAR1	000010	4	RW,D,CON,LCL
6	CBLAS	005120	1321	RW,D,OVR,GBL
7	DUMMY	000004	2	RW,D,OVR,GBL
8	CLOCK	000642	209	RW,D,OVR,GBL

ENTRY POINTS

NAME	TYPE	ADDRESS	NAME	TYPE	ADDRESS	NAME	TYPE	ADDRESS	NAME	TYPE	ADDRESS	NAME	TYPE	ADDRESS
CONRED		1-000000												

VARIABLES

NAME	TYPE	ADDRESS	NAME	TYPE	ADDRESS	NAME	TYPE	ADDRESS	NAME	TYPE	ADDRESS	NAME	TYPE	ADDRESS
BACATS	I*2	6-003316	BATOT	I*2	6-003766	BUCATS	I*2	6-001556	BUTOT	I*2	6-003770	CLADUM	I*2	7-000000
CLUDUM	I*2	7-000002	DF	I*2	6-001550	DJ	I*2	6-001552	I	I*2	6-003772	J	I*2	6-000000
J	I*2	4-000004	K	I*2	4-000006	PCOS1	R*4	6-003336	PCOS2	R*4	6-003340	PCO-	R*4	6-003326
PCO2	R*4	6-003332	PCLS1	R*4	6-003356	PCLS2	R*4	6-003342	PCL1	R*4	6-003346	PCL2	R*4	6-003352
TC	I*2	6-001554												

ARRAYS

NAME	TYPE	ADDRESS	SIZE		DIMENSIONS
A	R*4	6-001520	000030	12	(3,2)
BACAIN	R*4	6-000220	000400	128	(8,8)
BACLAB	L*1	6-003316	000010	4	(8)
BACOR	I*2	6-000140	000030	12	(12)
BALABL	L*1	6-001600	000010	4	(8)
BAPOP	I*2	6-000000	000030	12	(12)
BARANS	R*4	6-001350	000060	24	(12)
BAUNCJ	I*2	6-000060	000030	12	(12)
BAVAR	R*4	6-001220	000060	24	(12)
BCVA	I*2	6-003366	000200	64	(8,8)
BCVU	I*2	6-003566	000200	64	(8,8)
BUCAIN	R*4	6-000620	000400	128	(8,8)
BUCLAB	L*1	6-001560	000010	4	(8)
BUCOR	I*2	6-000170	000030	12	(12)
BULABL	L*1	6-001570	000010	4	(8)
BUPOP	I*2	6-000030	000030	12	(12)
BURANS	R*4	6-001440	000060	24	(12)
BUUNCJ	I*2	6-000110	000030	12	(12)
BUVAR	R*4	6-001300	000060	24	(12)
CORAS	R*4	6-005112	000010	4	(2)
CLAD	R*4	6-003772	000010	4	(2)
CLASFY	L*1	6-001610	000642	209	(418)

COMRED.FTN /TR:PL2C*S/LP

CLOSE	L01	6-000000	000542	209	(416)
CLUC	R04	6-004002	000010	4	(2)
LBLRD	L01	6-002773	000321	114	(205)
RCDLS	R04	6-004112	001000	256	(2,6,6)
TCBLS	R04	6-004012	000100	32	(2,8)
TYPE	L01	6-002652	000321	114	(209)

LABELS

LABEL	ADDRESS	LABEL	ADDRESS	LABEL	ADDRESS	LABEL	ADDRESS	LABEL	ADDRESS
9	**	10	**	100*	3-000004	101*	3-000007		

TOTAL SPACE ALLOCATED = 006346 1651

92 FPP INSTRUCTIONS GENERATED

COMRED,LP:=COMRED

```
OF77RAN IV-PLUS V02-51                    08:10:13    09-MAR-78          PAGE 1
MDTTL.FTN       /TRIBL7CKS/NR
0001            SUBROUTINE MDTTL(PTL)
0002            IMPLICIT INTEGER*2(A-Z)
0003            LOGICAL*1  MSD(12),PTL(1)
0004            DATA MSD/1HM,1HE,1HT,1HN,1H ,1H ,1HS,1HT,1H ,1HD,1HE,1HL/
0005            DO 1  I=1,12
0006     1      PTL(I)=MSD(I)
0007            RETURN
0008            END
```

PROGRAM SECTIONS

NUMBER	NAME	SIZE		ATTRIBUTES
1	$CODE1	000102	33	RW,I,CON,LCL
3	$IDATA	000012	5	RW,D,CON,LCL
4	$VARS	000016	7	RW,D,CON,LCL

ENTRY POINTS

NAME	TYPE	ADDRESS	NAME	TYPE	ADDRESS	NAME	TYPE	ADDRESS	NAME	TYPE	ADDRESS	NAME	TYPE	ADDRESS
MDTTL		1-000000												

VARIABLES

NAME	TYPE	ADDRESS	NAME	TYPE	ADDRESS	NAME	TYPE	ADDRESS	NAME	TYPE	ADDRESS	NAME	TYPE	ADDRESS
I	I*2	4-000014												

ARRAYS

NAME	TYPE	ADDRESS	SIZE		DIMENSIONS
MSG	L*1	4-000000	000014	6	(12)
PTL	L*1	F-000002*	000001	0	(1)

LABELS

LABEL	ADDRESS	LABEL	ADDRESS	LABEL	ADDRESS	LABEL	ADDRESS	LABEL	ADDRESS
1	**								

TOTAL SPACE ALLOCATED = 000132 45

NO FPP INSTRUCTIONS GENERATED

MDTTL.LP;*MDTTL

```
FNAME.FTN        /TR:BLOCKS/WR
0001             SUBROUTINE FNAME(FLD,PN)
0002             IMPLICIT INTEGER(A-Z)
0003             LOGICAL*1  FLD([illegible]),PN([illegible]),FIELD(6)
0004             DATA FIELD/1HF,1H-,1HE,1HL,1HD,1H /
0005             DO 11  II=1,6
0006             PN(II)=FIELD(II)
0007             PN(II+6)=FLD(II)
0008      11     CONTINUE
0009             RETURN
0010             END
```

```
FORTRAN IV-PLUS V02-54          [illegible]  15-MAR-77          PAGE 2
FNAME.FTN       /TR:BLOCKS/WD

PROGRAM SECTIONS

NUMBER   NAME      SIZE          ATTRIBUTES
  1      $CODE1   000132    45   RW,I,CON,LCL
  3      $IDATA   000024    10   RW,D,CON,LCL
  4      $VARS    000010     4   RW,D,CON,LCL


ENTRY POINTS

 NAME   TYPE  ADDRESS    NAME   TYPE  ADDRESS    NAME   TYPE  ADDRESS    NAME   TYPE  ADDRESS    NAME   TYPE  ADDRESS

 FNAME        1-000000


VARIABLES

 NAME   TYPE  ADDRESS    NAME   TYPE  ADDRESS    NAME   TYPE  ADDRESS    NAME   TYPE  ADDRESS    NAME   TYPE  ADDRESS

 II     I*2   4-000016


ARRAYS

 NAME   TYPE  ADDRESS      SIZE      DIMENSIONS

 FIELD  L*1   4-000000  000006    3   (6)
 FLD    L*1   F-000002* 000001    0   (1)
 PN     L*1   F-000004* 000001    0   (1)


LABELS

 LABEL  ADDRESS        LABEL  ADDRESS        LABEL  ADDRESS        LABEL  ADDRESS        LABEL  ADDRESS

 11        **


TOTAL SPACE ALLOCATED = 000166    59

NO FPP INSTRUCTIONS GENERATED

FNAME,LP1=FNAME
```

```
@FORTRAN IV-PLUS V02-51                    _CE1C6119     29-MAR-79          PAGE 1
KNT,FTN          /TR:BLOCKS/WR
0001          SUBROUTINE KNT
0002          IMPLICIT INTEGER(A-Z)
0003          COMMON/PCNT/LINE,PAGE
0004          LINE=LINE+1
0005          IF(LINE .LT. 60) GO TO 10
0006          PAGE=PAGE+1
0007          LINE=0
0008          WRITE(6,100) PAGE
0009       10 RETURN
0010      100 FORMAT('1',26X,'CAS/CAS INTERFACE TAPE PRINTOUT',14X,'PAGE',I4/)
0011          END
```

PROGRAM SECTIONS

NUMBER	NAME	SIZE	ATTRIBUTES
1	$CODE1	000076	31	PIC,I,LCL,LCL
3	$DATA	000052	25	PIC,D,LCL,LCL
6	PDT	000074	29	PIC,D,OVR,GBL

ENTRY POINTS

NAME	TYPE	ADDRESS	NAME	TYPE	ADDRESS	NAME	TYPE	ADDRESS	NAME	TYPE	ADDRESS	NAME	TYPE	ADDRESS
KNT		1-000000												

VARIABLES

NAME	TYPE	ADDRESS	NAME	TYPE	ADDRESS	NAME	TYPE	ADDRESS	NAME	TYPE	ADDRESS	NAME	TYPE	ADDRESS
LINE	I*2	6-000000	PAGE	I*2	6-000002									

LABELS

LABEL	ADDRESS	LABEL	ADDRESS	LABEL	ADDRESS	LABEL	ADDRESS	LABEL	ADDRESS
10	1-000074	100'	3-000000						

TOTAL SPACE ALLOCATED = 000164 58

NO FPP INSTRUCTIONS GENERATED

KNT,LP:=KNT

```
MEAN.FTN        /TR:BLOCKS/WR
                SUBROUTINE MEAN(FLD,PM,NCH,FLG)
                IMPLICIT INTEGER(A-Z)
                LOGICAL*1 FLD(1),PM(1),FLG(1),PT
                 LOGICAL*1 BLK
                DATA PT/1H./
                 DATA BLK/1H /
                OFF=11
                IF(FLG(1).EQ.'F')  OFF=12
                K=1
                 DO 3 J=1,NCH
                N=(J-1)*132
                 PM(N+1)=FLD(OFF+1)
                 PM(N+2)=FLD(OFF+2)
                 IF(PM(N+1).NE.'0')  GO TO 10
                 PM(N+1)=BLK
                 IF(PM(N+2).EQ.'0')  PM(N+2)=BLK
        10       PM(N+3)=FLD(OFF+3)
                 PM(N+4)=PT
                 PM(N+5)=FLD(OFF+4)
                 PM(N+6)=FLD(OFF+5)
                 PM(N+7)=BLK
                 PM(N+8)=FLD(OFF+6)
                 PM(N+9)=FLD(OFF+7)
                 IF(PM(N+8).NE.'0')  GO TO 11
                 PM(N+8)=BLK
                 IF(PM(N+9).EQ.'0')  PM(N+9)=BLK
        11       PM(N+10)=FLD(OFF+8)
                 PM(N+11)=PT
                 PM(N+12)=FLD(OFF+9)
                 OFF=OFF+9
         3      CONTINUE
                RETURN
                 END
```

MEAN,FTN /TR:RL3CMS/AP

PROGRAM SECTIONS

NUMBER	NAME	SIZE		ATTRIBUTES
1	$CODE1	000514	166	RW,I,CON,LCL
3	$IDATA	000036	15	RW,D,CON,LCL
4	.VARS	000012	5	RW,D,OVR,LCL
5	$TEMPS	000002	1	RW,D,CON,LCL

ENTRY POINTS

NAME	TYPE	ADDRESS	NAME	TYPE	ADDRESS	NAME	TYPE	ADDRESS	NAME	TYPE	ADDRESS	NAME	TYPE	ADDRESS
MEAN		1-000000												

VARIABLES

NAME	TYPE	ADDRESS	NAME	TYPE	ADDRESS	NAME	TYPE	ADDRESS	NAME	TYPE	ADDRESS	NAME	TYPE	ADDRESS
BLK	I*2	4-000000	J	I*2	4-000036	K	I*2	4-000034	L	I*2	4-000032	M	I*2	5-000000
ZZZ	I*2	4-000012	PI	R*4	4-000030									

ARRAYS

NAME	TYPE	ADDRESS	SIZE		DIMENSIONS
FLG	L*1	F-000002*	000001	0	(1)
FLG	L*1	F-000010*	000001	0	(1)
PM	L*1	F-000004*	000001	0	(1)

LABELS

LABEL	ADDRESS	LABEL	ADDRESS	LABEL	ADDRESS	LABEL	ADDRESS	LABEL	ADDRESS
3	**	10	1-000260	11	1-000427				

TOTAL SPACE ALLOCATED = 000546 187

NO FPP INSTRUCTIONS GENERATED

MEAN,LP:=MEAN

```
@F2RTRAN IV-PLUS V02-54                    08105125    09-MAR-75          PAGE 1
P2P.FTN          /TR:PROCS/WR
0001             SUBROUTINE P2P(FLG,PP,FLG)
0002             IMPLICIT INTEGER(A-Z)
0003             LOGICAL*1 FLG(1),FLG(2),PAR(2),FLG(1)
0004             DATA PAR/2H(,2H)/
0005             BFF=6
0006             IF(FLG(1).EQ.'F')  BFF=5
0007             PP(3)=PAR(1)
0008             DO  1  I=1,5
0009             PP(I+3)=FLG(I+BFF)
0010     1       CONTINUE
0011             PP(9)=PAR(2)
0012             RETURN
0013             END
```

```
PROGRAM SECTIONS

NUMBER   NAME       SIZE               ATTRIBUTES

  1      $CODE1   000216    71         RW,I,CON,LCL
  3      $IDATA   000036    15         RW,D,CON,LCL
  4      $VARS    000006     3         RW,D,CON,LCL


ENTRY POINTS

 NAME   TYPE  ADDRESS    NAME  TYPE  ADDRESS    NAME  TYPE  ADDRESS    NAME  TYPE  ADDRESS    NAME  TYPE  ADDRESS

 P2P          1-000000


VARIABLES

 NAME   TYPE  ADDRESS    NAME  TYPE  ADDRESS    NAME  TYPE  ADDRESS    NAME  TYPE  ADDRESS    NAME  TYPE  ADDRESS

 I      I*2   4-000004   OFF   I*2   4-000002


ARRAYS

 NAME   TYPE  ADDRESS        SIZE          DIMENSIONS

 FLD    L*1   F-000002*  000001     0      (1)
 FLG    L*1   F-000006*  000001     0      (1)
 PAR    L*1   4-000000   000002     1      (2)
 PP     L*1   F-000004*  000001     0      (1)


LABELS

 LABEL   ADDRESS        LABEL   ADDRESS        LABEL   ADDRESS        LABEL   ADDRESS        LABEL   ADDRESS

 1         **


TOTAL SPACE ALLOCATED = 000262    89

NO FPP INSTRUCTIONS GENERATED

P2P,LP:=P2P
```

```
      SUBROUTINE SNAME(FLD,PN,FLG)
      IMPLICIT INTEGER(A-Z)
      LOGICAL*1 FLD(1),PN(1),SUBCL(6),FLG(1)
      DATA SUBCL/1HS,1H ,1HR,1HC,1H-,1H /
      OFF=2
      IF(FLG(1).EQ.'F') OFF=7
      DO 11 II=1,6
      PN(II)=SUBCL(II)
      PN(II+6)=FLD(II+OFF)
   11 CONTINUE
      RETURN
      END
```

```
FORTRAN IV-PLUS V02-51          10:06:32    09-MAR-78          PAGE 2
SNAME.FTN          /TR:BLOCKS/WR
PROGRAM SECTIONS

NUMBER   NAME        SIZE              ATTRIBUTES

  1     $CODE1  000252    85          RW,I,CON,LCL
  3     $DATA   000036    15          RW,D,CON,LCL
  4     $VARS   000012     5          RW,D,CON,LCL


ENTRY POINTS

 NAME   TYPE  ADDRESS     NAME   TYPE  ADDRESS     NAME   TYPE  ADDRESS     NAME   TYPE  ADDRESS     NAME   TYPE  ADDRESS

 SNAME        4-000000


VARIABLES

 NAME   TYPE  ADDRESS     NAME   TYPE  ADDRESS     NAME   TYPE  ADDRESS     NAME   TYPE  ADDRESS     NAME   TYPE  ADDRESS

 II     I*2   4-000010    OFF    I*2   4-000006


ARRAYS

 NAME   TYPE  ADDRESS            SIZE         DIMENSIONS

 FLD    L*1   F-000002* 000001     0         (1)
 FLS    L*1   F-000006* 000001     0         (1)
 PN     L*1   F-000024* 000001     0         (1)
 SUBCL  L*1   4-000000  000006     3         (6)


LABELS

 LABEL  ADDRESS           LABEL  ADDRESS           LABEL  ADDRESS           LABEL  ADDRESS           LABEL  ADDRESS

 11         **


TOTAL SPACE ALLOCATED = 000252    85

NO FPP INSTRUCTIONS GENERATED

SNAME,LP:=SNAME
```

```
STRIP.FTN        /TR:BLOCKS/WR
                 SUBROUTINE STRIP(IBUF)
                 IMPLICIT INTEGER(A-Z)
                 LOGICAL*1 IBUF(1)
                 COMMON /PORT/LINE,PAGE
                 LINE=66
                 CALL PNT
                 IF(IBUF(2).EQ.'F')  GO TO 8
                 WRITE(6,200)
      200        FORMAT(5X,'SUBCLASS STATISTICS RECORD')
                 GO TO 9
      8          WRITE(6,201)
      201        FORMAT(5X 'FIELD STATISTICS RECORD')
      9          CONTINUE
                 WRITE(6,202)  (IBUF(N),N=3,4)
      202        FORMAT(5X,'RECORD SEQ. NO.  ',2A1)
                 WRITE(6,203)  (IBUF(N),N=5,7)
      203        FORMAT(5X,'NO. OF SUBCLASSES  ',3A1)
                 WRITE(6,204)  (IBUF(N),N=8,9)
      204        FORMAT(5X,'NO. OF CLASS SETS ',2A1)
                 WRITE(6,205)  (IBUF(N),N=10,11)
      205        FORMAT(5X,'NO. OF CHANNELS   ',2A1)
                 LINE=5
                 RETURN
                 END
```

```
FORTRAN IV-PLUS V02-51          15:06:37    05-MAR-78          PAGE 2
STDMP.FTN        /TR:BLOCKS/WR
PROGRAM SECTIONS

NUMBER  NAME       SIZE                   ATTRIBUTES

  1     $CODE1   000460   152             RW,I,CON,LCL
  3     $IDATA   000256    87             RW,D,CON,LCL
  4     $VARS    000002     1             RW,D,CON,LCL
  6     PCNT     000004     2             RW,D,OVR,GBL

ENTRY POINTS

 NAME   TYPE  ADDRESS    NAME   TYPE  ADDRESS    NAME   TYPE  ADDRESS    NAME   TYPE  ADDRESS    NAME   TYPE  ADDRESS

 STDMP        1-000000

VARIABLES

 NAME   TYPE  ADDRESS    NAME   TYPE  ADDRESS    NAME   TYPE  ADDRESS    NAME   TYPE  ADDRESS    NAME   TYPE  ADDRESS

 LINE   I*2   6-000000   N      I*2   4-000000   PAGE   I*2   6-000002

ARRAYS

 NAME   TYPE  ADDRESS          SIZE         DIMENSIONS

 IBUF   L*1   F-000002* 000001    9         (1)

LABELS

 LABEL  ADDRESS          LABEL  ADDRESS          LABEL  ADDRESS          LABEL  ADDRESS          LABEL  ADDRESS

 8      1-000120         9      1-000146         200'   3-000000         201'   3-000040         202'   3-000074
 203'   3-000126         204'   3-000156         205'   3-000210

FUNCTIONS AND SUBROUTINES REFERENCED

 KNT

TOTAL SPACE ALLOCATED = 000744   242

NO FPP INSTRUCTIONS GENERATED

STDMP,LP:=STDMP
```

ORIGINAL PAGE IS OF POOR QUALITY

```
FORTRAN IV-PLUS V02-51              08:05:42    09-MAR-76          PAGE  1
PRT.FTN          /TRUNCATE/CHECK/WARN
0001              SUBROUTINE PRT
0002              IMPLICIT INTEGER(A-Z)
0003              COMMON/PCNT/LINE,PAGE
0004              LINE=LINE+1
0005              IF(LINE .LT. 59) GO TO 1
0006              PAGE=PAGE+1
0007              LINE=0
0008              WRITE(3,100)
0009            1 RETURN
0010          100 FORMAT('1',71X,'    ')
0011              END
```

```
FORTRAN IV-PLUS V02-54          09106142    05-MAR-78          PAGE 2
BNT,FTN          /TR:RLOCKS/RE
PROGRAM SECTIONS

  NUMBER    NAME        SIZE                ATTRIBUTES

    1       $CODE1   000066    27          RW,I,CON,LCL
    3       $IDATA   000014     6          RW,D,CON,LCL
    6       PCNT     000004     2          RW,D,OVR,GBL

ENTRY POINTS

  NAME   TYPE  ADDRESS    NAME   TYPE  ADDRESS    NAME   TYPE  ADDRESS    NAME   TYPE  ADDRESS    NAME   TYPE  ADDRESS

  BNT           1-000000

VARIABLES

  NAME   TYPE  ADDRESS    NAME   TYPE  ADDRESS    NAME   TYPE  ADDRESS    NAME   TYPE  ADDRESS    NAME   TYPE  ADDRESS

  LINE   I*2   6-000000   PAGE   I*2   6-000002

LABELS

  LABEL   ADDRESS         LABEL   ADDRESS         LABEL   ADDRESS         LABEL   ADDRESS         LABEL   ADDRESS

  10      1-000064        100     3-000000

TOTAL SPACE ALLOCATED = 000106   35

 0 FPP INSTRUCTIONS GENERATED

BNT,LP:=BNT
```

```
*FORTRAN IV-PLUS V02-51                    18:06:46    19-MAR-79          PAGE 1
CRIP2.FTN          /TR:HL2CRS/.R
0001               SUBROUTINE CRIP2(FLD,CLASS,P1,P2)
0002               IMPLICIT INTEGER*2(A-Z)
0003               LOGICAL*1 FLD(1),CLASS(1)
0004               DO 2 I=1,4
0005               CLASS(I)=FLD(I)
0006     2         CONTINUE
0007               DECODE(5,100,FLD(5)) P1
0008     100       FORMAT(I5)
0009               DECODE(5,100,FLD(9)) P2
0010               RETURN
0011               END
```

```
FORTRA  IV-PLUS V02-51          [illegible]     09-MAR-79          PAGE 2
CPIP2.FTN      /TR:BLOCKS/WR
PROGRAM SECTIONS
$
NUMBER   NAME       SIZE                  ATTRIBUTES
$
  1      $CODE1  000224    74             RW,I,CON,LCL
  3      $IDATA  000030    12             RW,D,CON,LCL
  4      $VARS   000002     1             RW,D,CON,LCL
$
$
ENTRY POINTS
$
 NAME    TYPE  ADDRESS     NAME   TYPE  ADDRESS     NAME   TYPE  ADDRESS     NAME   TYPE  ADDRESS     NAME   TYPE  ADDRESS
$
 CPIP2         1-000000
$
$
VARIABLES
$
 NAME    TYPE  ADDRESS     NAME   TYPE  ADDRESS     NAME   TYPE  ADDRESS     NAME   TYPE  ADDRESS     NAME   TYPE  ADDRESS
$
 I       I*2   4-000000    PI     I*2   F-000036*   PX     I*2   F-000030*
$
$
ARRAYS
$
 NAME    TYPE  ADDRESS         SIZE             DIMENSIONS
$
 CLASS   L*1   F-000004* 000001      0    (1)
 FLD     L*1   F-000002* 000001      0    (1)
$
$
LABELS
$
 LABEL    ADDRESS            LABEL    ADDRESS            LABEL    ADDRESS            LABEL    ADDRESS            LABEL    ADDRESS
$
 2             **            1001     3-000000
$
$
TOTAL SPACE ALLOCATED = 000256    87
$
NO FPP INSTRUCTIONS GENERATED
$
CPIP2,LP:=CPIP2
```

```
$FORTRAN IV-PLUS V02-51                  08:05:51    09-MAR-78          PAGE 1
MV.FTN          /TR:BLOCKS/WP
0001            SUBROUTINE MV(FLD,PB,NC)
0002            IMPLICIT INTEGER(A-Z)
0003            LOGICAL*1 FLD(1),PB(1)
0004            DO  8  J=1,NC
0005        8   PB(J)=FLD(J)
0006            RETURN
0007            END
```

MV.FTN /TR:SLBCKS/AS

PROGRAM SECTIONS

NUMBER	NAME	SIZE	ATTRIBUTES
1	$CODE1	000124	42	RW,I,CON,LCL
3	$IDATA	000024	10	RW,D,CON,LCL
4	$VARS	000002	1	RW,D,OVR,LCL
5	$TEMPS	000002	1	RW,D,OVR,LCL

ENTRY POINTS

NAME	TYPE	ADDRESS	NAME	TYPE	ADDRESS	NAME	TYPE	ADDRESS	NAME	TYPE	ADDRESS	NAME	TYPE	ADDRESS
MV		1-000000												

VARIABLES

NAME	TYPE	ADDRESS	NAME	TYPE	ADDRESS	NAME	TYPE	ADDRESS	NAME	TYPE	ADDRESS	NAME	TYPE	ADDRES
J	I*2	4-000000	N	I*2	F-000000*									

ARRAYS

NAME	TYPE	ADDRESS	SIZE	DIMENSIONS
FLD	L*1	F-000002*	000001	3	(1)
PR	L*1	F-000003*	000001	3	(1)

LABEL

LABEL	ADDRESS	LABEL	ADDRESS	LABEL	ADDRESS	LABEL	ADDRESS	LABEL	ADDRESS
8	**								

TOTAL SPACE ALLOCATED = 000154 54

NO FPP INSTRUCTIONS GENERATED

APPENDIX B

The values Dw and Ds are computed as follows:

$Dw = C_1 + \sum_{p=1}^{8} \sum_{q=1}^{8} RpqWpWq + \sum_{p=1}^{8} TpWp$ or 0 if Category W not used

$Ds = C_2 + \sum_{p=1}^{8} \sum_{q=1}^{8} UpqSpSq + \sum_{p=1}^{8} VpSp$ or 0 if Category S not used

where C_1, C_2, $\{Rpq\}_{p=1\ q=1}^{8\ 8}$, $\{Tp\}_{p=1}^{8}$, $\{Upq\}_{p=1\ q=1}^{8\ 8}$, and $\{Vp\}_{p=1}^{8}$ are constants, supplied by user

W_1 = Bias corrected estimate for Category W

S_1 = Bias corrected estimate for Category S

W_2 = Machine estimate for Category W

S_2 = Machine estimate for Category S

W_3 = Random estimate for Category W

S_3 = Random estimate for Category S

W_4 = Variance of Bias corrected estimate of W

S_4 = Variance of Bias corrected estimate of S

$$W_5 = \frac{W:W}{W:W + S:W + N:W}$$

$$S_5 = \frac{S:S}{W:S + S:S + N:S}$$

$W_6 = S_6 = PCC1$

$$W_7 = S_7 = \frac{N:N}{W:N + S:N + N:N}$$

$W_8 = S_8 = PCC2$

APPENDIX C

a. Constants for Dw calculations

Card 1 contains C_1 as follows:
blanks or XXX.XXXXX in first 10 columns only

Card 2 contains T_1 thru T_8 as follows:
blanks or XXX.XXXXX for each entry. A maximum of 10 columns wide starting in columns 1, 11, 21, etc.

Card 3 thru 10 contain $R_{1,1}$ to $R_{8,8}$. $R_{1,1}$ thru $R_{1,8}$ on card 3, $R_{2,1}$ thru $R_{2,8}$ on card 4, etc. Format same as card 2.

b. Constants for Ds calculations

Card 1 contains C_2 as follows:
blanks or XXX.XXXXX in first 10 columns only

Card 2 contains V_1 thru V_8 as follows:
blanks or XXX.XXXXX for each entry. A maximum of 10 columns wide starting in columns 1, 11, 21, etc.

Card 3 thru 10 contain $U_{1,1}$ to $U_{8,8}$. $U_{1,1}$ thru $U_{1,8}$ on card 3, $U_{2,1}$ thru $U_{2,8}$ on card 4, etc. Format same as card 2.

APPENDIX D

Change 1
July 31, 1978