DEPARTMENT OF
MINERALS AND ENERGY

BUREAU OF MINERAL RESOURCES, GEOLOGY AND GEOPHYSICS

Record 1973/211

THE STORAGE AND RETRIEVAL SYSTEM FOR HYDROGEOLOGICAL DATA FROM THE GREAT ARTESIAN BASIN

by

G.E. Seidel

Record 1973/211

THE STORAGE AND RETRIEVAL SYSTEM FOR HYDROGEOLOGICAL DATA FROM THE GREAT ARTESIAN BASIN

by

G.E. Seidel

CONTENTS

FIGURES

SUMMARY

The original form of storage for the hydrogeological data from the Great Artesian Basin and for the drillstem test data is an ordered set of card decks. Large data files based on punched cards are bulky and vulnerable. The data can be stored safely, and handling can be simplified by basing the processing on copies of the data sets held on magnetic tape or disc. A storage and retrieval system based on magnetic tape and disc was developed.

Data are grouped into a hierarchy of subdivisions called Volume, Page, and Line. The system stores and retrieves data in units of Pages, and maintains a catalogue listing the number of Lines to each Page, and Pages to each Volume. Different storage levels are provided for input/editing, temporary storage (disc), and permanent storage (tape). Retrieved data are stored as separate disc files, which can be read like standard card decks. The contents of up to 200 000 punched cards can be stored on one tape.

A major design consideration was simplicity of operation and very little data processing experience is required of the user. The system was developed for the CYBER 76 computer of the CSIRO Division of Computing Research, Canberra.

1. INTRODUCTION

1.1 Scope of the System

This Record follows that of Ungemach & Habermehl (1973), which describes the scope of the Great Artesian Basin (GAB) Automatic Data Processing System and outlines Stage 1 of this system (Data Transcription), and (Krebs, 1973), which describes Stage 2, the checking of coded data. The subject of this record is the permanent storage, updating, and retrieval for processing of the data passed through Stages 1 and 2. The system described was developed for application to drill stem test (DST; Formation Test) data by G.E. Seidel (BMR) and then extended to suit the general GAB data by G. Krebs (BRGM).

1.2 Purpose of the System

Whilst passing through Stages 1 and 2 of the ADP system, data are coded, punched on cards, crosschecked and stored as a set of card decks. Although well organized these card decks are bulky and vulnerable to damage from card readers or disarrangement through handling errors. They form a valuable investment particularly in time spent during the punching, verifying, sorting, and checking operations, and consequently should be protected as much as possible.

A solution to protect the original data, and, at the same time, eliminate the bulk to be handled during processing is to base all processing on magnetic tape or disc copies of the original data. Selection of the storage media depends on the application. Generally, tape is preferred for long-term storage, and disc for processing.

The purpose of the utility system, described in this record, is to produce working copies of the original card data file on magnetic disc, tape, or both, in forms suitable for long-term storage, and to provide for partial or complete retrieval of the stored data for processing purposes.

1.3 Design Considerations

Requirements the utility system must meet are:

- Input of the data must be step by step and unit by unit, for it is physically impossible to submit the whole card file for processing simultaneously.
- A facility must be provided to insert corrections or additional data and to delete faulty or redundant data.
- Retrieval must be possible in a variety of forms to meet the requirements of processing programs.

Additional operational characteristics which should be included for efficiency of operation and for convenience are:

- The correct sorting of cards should be re-checked during input before writing on tape.
- The number of tape-writing operations should be kept to a minimum.
- For rapid access, individual editing or updating procedures should be based on disc or drum files rather than magnetic tape.
- The system should provide an up-to-date summary or catalogue of the data stored.
- Specification of data selected for retrieval should be simple.
- The system should be controllable by remote-console terminal as well as by program card decks.
- Operation of the system should not require specialized data processing experience.
- The system should be of a design transferrable between different computers without principal modifications.

A balance between fully providing all of these features and the required programming effort may have to be found.

1.4 Existing Data Storage and Retrieval Systems

A variety of useful software packages is available from some organizations, so their use naturally had to be considered. Generally these systems have been designed to cope with most types of data and they usually have provision for editing by insertion of correction cards (CARDIMED by CSIRO, UPDATE by CDC). However, their generality introduces an overhead of unused options when applied to specialized and highly systematic data sets. On the other hand, useful options which can be developed for a specialized data set could not be included in such a system without sacrificing its generality. As a consequence these standard systems find their main use in storage of non-homogeneous data files or as temporary solutions. Whenever data files are large, specialized, and systematic, and frequent use is anticipated, it becomes likely that the programming effort in implementing a specialized system is justified by the gain in the efficiency of operation.

The data collected in the GAB card file fit this description and the feasibility of implementing a specialized system was studied. As a result it was found that a system meeting all requirements listed (see 1.3 Design Considerations) could be produced with relatively little programming making use of the inherent logic of the data set.

1.5 Notes on the Presentation

The system is applied to two different data sets, the general GA data set, and the DST data set, which represents data from drill-stem testing of petroleum exploration wells within the Great Artesian Basin. Both data sets are of similar design but differ in some aspects and size, and they require minor differences in programs. The program logic presented applies to both systems, and corresponding program and file names differ by their prefix only, e.g. GCARME, GABIN for the GAB system correspond to DCARME, DSTIN in the DST system.

The system was tried out on the DST data first. Flow charts, program variables, and sample print-outs are those of the DST system except where stated otherwise. Differences between the systems are indicated where relevant.

The original version was written for the CDC 3600 computer of the CSIRO computing centre in Canberra. However the CDC 3600 is being phased out and replaced by the much faster CYBER 76. The presented version has been written for the CYBER 76. Interactive editing through terminals is not yet possible on the CYBER 76, but data sets and programs can be communicated between the machines. The present system is a hybrid with terminal editing performed through the 3600, and processing based on the CYBER 76.

2. DESCRIPTION OF THE SYSTEM

2.1 Grouping of Data

To avoid confusion with the various terms for data organization defined differently on different computer installations, a separate terminology is used for the subdivision of the data set to be handled. These subdivisions can then be equated to the terms applicable to the particular computer installation used.

- All available data on punched cards from the GAB or DST system form the GAB or DST Data Set.

- The first subdivision of a data set is referred to as a Volume and contains all data referring to wells on a particular map sheet (GAB) or wells completed during a particular year (DST).

- The Volume is subdivided into units called Pages, each of which contains all data punched on the same type of card in the original card version of the data set.

- The smallest unit, the Line, is equivalent to an individual card.

The original card data file for both the GAB and DST data sets are organized in this manner (Fig. 1). How the subdivisions are equated to computer-installation-defined terms depends on hardware and software characteristics of the computer installation used. For example in the original version applied to the CDC 3600 a Line was equivalent to a Physical Record and a Page to a File. The Page was the basic transfer unit between storage devices to utilize efficient copy utilities during tape-read-and-write operations.

The CYBER 76 allows the efficient copying of whole tapes to and from system disc (staging); this eliminates the need for the transfer of larger units between storage media except for the staging operation itself. Hence, the basic unit for the system on the CYBER 76 is the Line. Control over larger subdivisions is still maintained by means of the catalogues (see 2.4 Data Catalogues).

2.2 Storage Levels and Editions

The system employs three separate levels of temporary or permanent storage. The movement of data through these levels during the input sequence is illustrated in Figure 2.

Level 1 is for short-term storage during the first stage of the input sequence. Its purpose is to allow the final checking, and if necessary, editing of data. Interactive editing through a terminal is not possible on data sets stored on the CYBER 76 disc, and programs on the CYBER 76 cannot make direct access of data stored on the document region (magnetic drum) of the CDC 3600. However data files can be sent (disposed) from the CYBER to the 3600, and programs plus data can be sent to the CYBER for processing. The level 1 data file and catalogue is created on the CYBER and a copy is sent to the 3600. This copy can be edited and resubmitted together with a copy of the input program to the CYBER. The resultant new edition of the level 1 file replaces the old one.

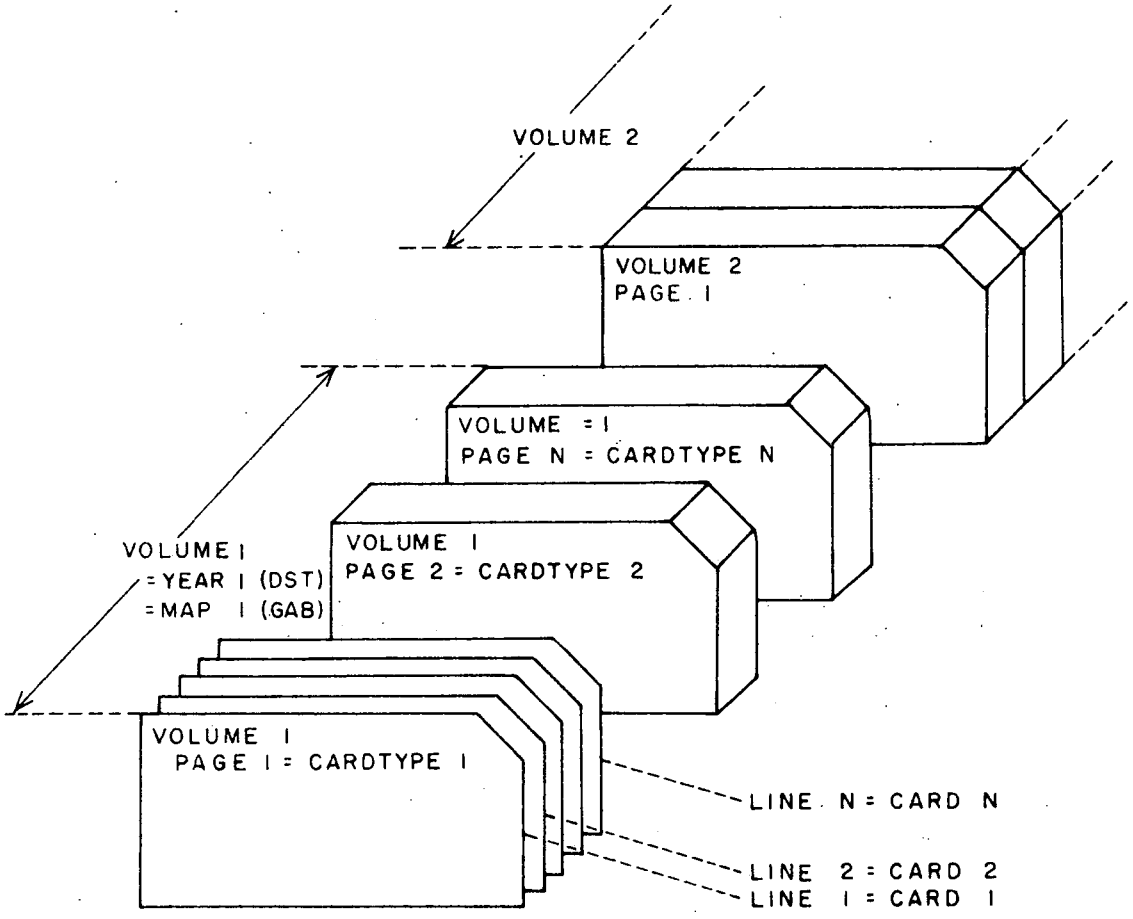


Fig. 1 Grouping of data GAB and DST data sets.

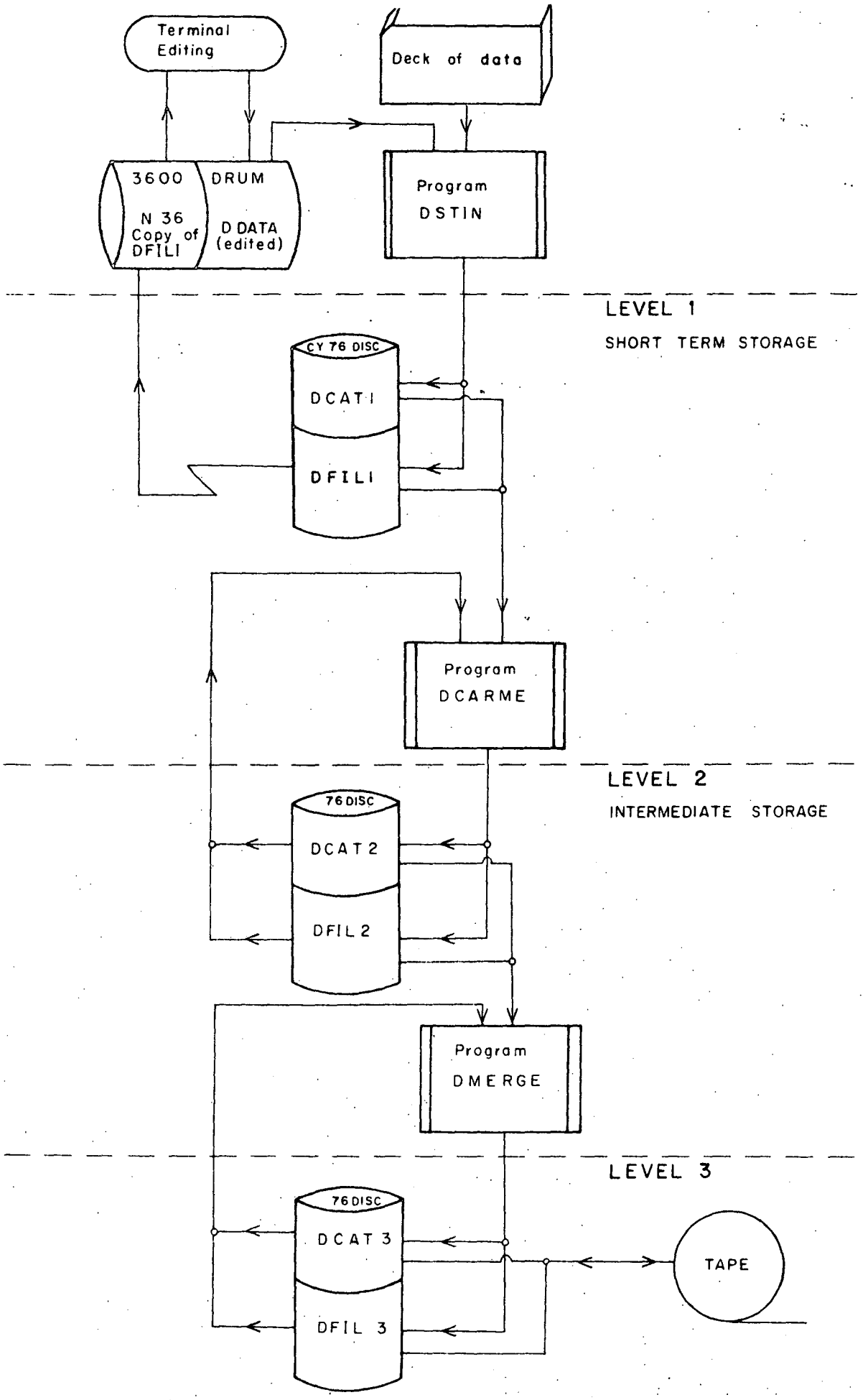


Fig. 2 Movement of data through storage levels.

Level 2 is an intermediate storage level and resides completely on the CYBER 76 system disc. Its purpose is to accumulate completed data files from level 1 until the size of the accumulated file, or processing requirements, warrant rewriting (updating) of the permanent level 3 file.

Level 3, the permanent file, resides on magnetic tape. It is updated by merging the current-edition 1 tape with the current-level 2 file to create the new-edition 1 tape. A copy of the level 2 file is retained separately to allow the repetition of the merging process should the new tape be destroyed or its contents interfered with.

There are always at least two tape editions retained. The current tape is edition 1. This edition is used for processing and is the input tape when a new edition is to be written by merging with the level 2 file. The new tape initially is called edition 0 until it has been established that the merging and tapewrite process was successful. It then becomes the new-edition 1 tape, the old-edition 1 tape is renamed edition 2 or backup tape, and the old-edition 2 tape is released.

2.3 Data Formats

The original data sets consist of standard 80-column punched cards. The format of these cards is retained throughout the system in the form of card images or logical records.

The information on the cards is treated as alphanumeric data; no attempt to decode is made except by the input program (GABIN, DSTIN), which in order to check the sorting and to establish the catalogue of data, reads the variables used for sorting the data set.

- Volume number (DST: year number, GAB: map sheet numbers)
- Page (Card type number)
- Well number (note: several Lines may refer to the same well)

During the checking and cataloguing operation the input program assigns a consecutive number to each Line (card image) to facilitate the terminal editing of the level 1 file generated by the program.

The transfer from level 1 to level 2 is performed without decoding. The catalogues and a count of Lines are used to ensure the proper merging into the level 2 file. Lines are renumbered during the merging process and again consecutive Line numbers are prefixed to the card images.

The consecutive Line numbers are stripped off during the transfer of data to the level 3 file. The level 3 file is written first as a 'tape image' on disc. Blocks (physical records) larger than the logical record (Line) are used to preserve storage space. In the system implemented on the CYBER each Line is in the form of a control-word-type (W type) record; blocking is of the internal type (I type) defined with a maximum block length of 5120 characters. Logical records may span adjoining blocks. With this arrangement and a tape density set to 800 characters per inch up to 200 000 card images can be stored on a single tape.

To conserve time the internal binary 'Display Code' of data is retained. Before staging the tape-image file out onto magnetic tape, a standard (ANSII) tape label is written onto the tape. The resulting tape can easily be copied onto a new tape in any standard interchange format should this become necessary.

2.4 Data Catalogues

The data catalogue is the main operating tool of the system. It is established during the input sequence. The input program, whilst checking the proper sorting of data cards, counts the number of Lines for each Page of a particular Volume. Each line of the catalogue consists of the Volume number (and optionally map sheet designation) followed by a list giving the number of Lines for each Page (see program printout in APPENDIX B for example). If a Page contains no Lines a zero will be written in the catalogue at the appropriate position. However, an empty Volume does not appear in the catalogue at all.

Each storage level has its own catalogue, DCAT1 or GCAT1 on level 1, DCAT2 or GCAT2 on level 2, etc. During merging operations, e.g. of level 1 with level 2, both catalogues are compared. If, for example, level 1 contains only Page N of Volume M and there are L Lines on this Page, then the merging program will copy onto the new level 2 file all data proceeding Page N of Volume M from the old level 2 file, then L Lines from the level 1 file, and then all data following Page N of Volume M again from the old level 2 file. At the same time L will be inserted at position N for Volume M in the new level 2 catalogue.

On the permanent level 3 file (magnetic tape) the catalogue is written first, followed by an End of File (End of Partition) mark, then the data. This catalogue always provides an accurate summary of the contents of the tape and a count of the number of Lines to be skipped to reach any particular Line to be retrieved.

2.5 Storing Sequence

Before the data can be stored, they must have passed through stages 1 and 2, (i.e. they have been coded, punched on cards, and the cards have been checked for coding and punching errors). If they are not yet sorted according to Volume, Page, and Well number, the sorting has to be carried out first. Volumes need not be complete but Pages are treated as complete units.

A card deck of manageable size containing one or more completed Pages is assembled and submitted with program DSTIN (GABIN). The program checks the sorting of the cards, counts the number of Lines per Page, and compiles the catalogue of the submitted data (see 2.4 Data Catalogues). Cards out of sequence are indicated under Error Listing in the program printout and are not included in the Line count of the catalogue. All input Lines are given a consecutive line number to which any error messages refer. The catalogue and the input data are stored on the CYBER 76 disc as level 1 files, a copy of the data file is sent to the drum of the 3600, where it is accessible to editing through the terminal network (see Fig. 2).

Any editing is performed at this stage either by correcting the original input card deck and resubmitting it with the input program, or by editing the copy on the 3600 drum and resubmitting this with a special copy of the input program, which is stored on drum permanently.

As soon as the level 1 files are found to be correct they are transferred by submitting program DCARME (GCARME) to level 2. Program DCARME compares the catalogue of the level 1 file with the catalogue of the level 2 file (if existing) and merges the two files Page by Page into the new level 2 file and catalogue. New editions of a particular Page will overwrite older versions existing already on level 2. After the merging both level 1 file and catalogue are cleared to make room for new entries by the input program.

Once sufficient data have accumulated on the level 2 file to warrant rewriting of the level 3 file (magnetic tape) program DMERGE (GMERGE) is submitted. This program is very similar to DCARME. By comparing the respective catalogues it merges the level 2 file with the current level 3 file, so creating a new level 3 catalogue and file, which are stored on a new tape. The old tape is retained as a safety measure and for the same reason a copy of the level 2 file is stored as a separate file before the level 2 file is cleared. This allows a repetition of the merging operation in case the writing of the new tape was affected by a system

breakdown. The new tape becomes the current tape storing both the level 3 catalogue and file.

2.6 Retrieval Sequences

Retrieval is from magnetic tape and is possible either partial or complete on a Page by Page basis. Retrieved data are stored on one or more disc files on the CYBER 76 where they can be accessed by processing programs (Fig. 3).

Retrieval is by program DRETRI (GRETRI) which reads one or more cards with formatted retrieval requests followed by two cards specifying on what disc files retrieved pages are to be stored and whether or not different pages are to be separated by End of Partition (EØP or EØF) marks on their disc files.

The current magnetic tape is copied onto the system disc in full (staging). The resulting file is split into level 3 catalogue and file by copying onto unblocked disc files. The specifications in the retrieval requests, which must be in ascending order, are compared with the catalogue.

If a specified Page of a specified Volume is found, its position in the catalogue serves to locate it on the level 3 file. It is then copied onto the disc file specified for retrieval of that Page. If it is not found on the catalogue a message to that effect appears on the printout. The disc files containing the retrieved data are in unblocked card-image form and can be read by processing programs like card decks.

2.7 Updating Sequence

Updating is carried out as a combination of retrieval and input. A specialized version of the retrieval program (DTAPEX, GTAPEX) extracts the specified Pages from the level 3 file and copies them all onto disc file with no intervening EØP marks.

It then activates a copy of the input program, which reads and recatalogues the retrieved data and sends a copy of the generated level 1 file to the 3600 drum. There the data are accessible to editing through the terminal network. After completion of the editing, the data are re-introduced through the normal storing sequence replacing older versions of corresponding data on tape.

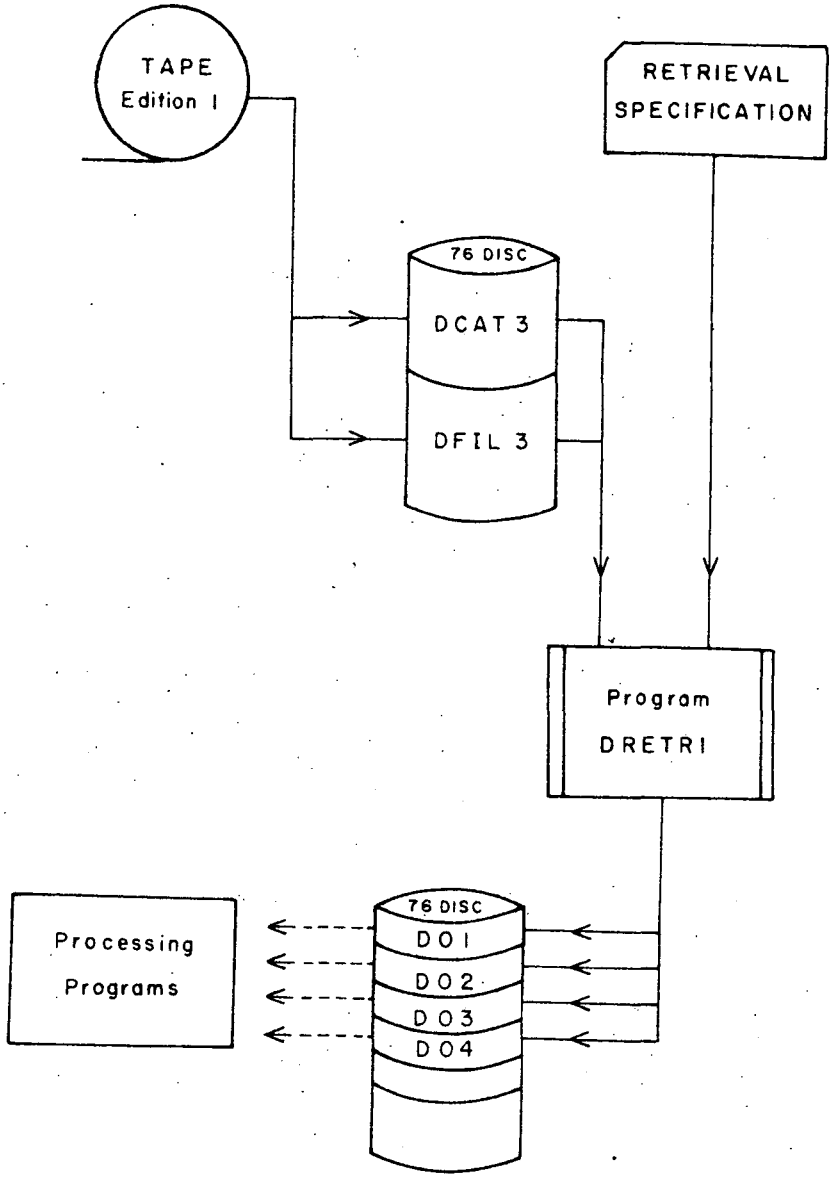


Fig. 3 Retrieval sequence

Alternatively, updating may be carried out by simply including the card decks of updated Pages into the normal storing sequence.

3. DESCRIPTION OF SYSTEM PROGRAMS

3.1 Input Programs (DSTIN, GABIN)

Three versions exist of the input programs. Version 1 is used with a deck of cards as input data, and is the basic version. Input for version 2 is card images preceeded by Line numbers stored as document on the 3600 drum. This version is used for reprocessing edited input data and differs from version 1 in one FORMAT statement only.

Version 3 is part of the updating routine and instead of a card deck it reads a disc file on the CYBER 76 created by DTAPEX (GTAPEX), a special version of the retrieval program. It differs from version 1 in one file access statement (ATTACH....) and in a corresponding change in the PROGRAM statement.

Job control statements for version 1 and 2 are shown in Figure 4. The FORTRAN program is compiled, executed, and the resulting level 1 files DCAT1 and DFIL1 are 'cataloged' as permanent files. An error results if the level 1 files are occupied already. This error occurs normally whenever corrected data are resubmitted with the input program. The old level 1 files are then removed (PURGE) and a second attempt is made to catalogue the new level 1 files. Finally a copy of the new level 1 data file is sent to the 3600 drum document region.

Detailed flowcharts and an explanation of variables for program DSTIN are presented in Figures 5 and 6.

Values for previous Volume, Page, Well number (NPY, NPC, NPW), line number, and control variable (IR, LOG) are initialized to zero. Then the first card is read decoding Page number, Volume number, and Well number; the remainder are left as undecoded alphanumeric data.

Following the reading of any card the Volume number is compared with the previous Volume number. If the Volume number is less than the previous Volume number (NPY-NY 0) the cards are out of sequence. An appropriate message is printed and the card is included in the level 1 data file but is not added to the count for the catalogue.

If the Volume numbers are equal (NPY-NY = 0), the card just read belongs to the same Volume as the previous one, and the Page numbers are compared next. If the Volume number is greater than the previous one (NPY-NY 0) a new Volume is started. Then the catalogue line for the previous Volume is complete and is written on the level 1 catalogue file - except if LOG = 0, i.e. a previous Volume does not exist. If the Volume number is 99 the end of the data set has been reached; otherwise the previous Volume number is reset to the new one, and the previous Page, the well number, and the Line count for the catalogue are reset to zero.

The Page numbers are then compared. Wrong sorting has the same effect as already described for the Volume numbers. If the Page numbers match, the Well numbers are compared next. If the new Page number is greater than the previous one a new Page is started. The previous Page number is reset to the new one, and the previous well number is reset to zero.

Finally the well numbers are compared. If the well number is equal to or greater than the previous one, sorting is correct and the count of Lines for the current page is incremented by one. Otherwise a message is printed. The consecutive line member is incremented by one and this line number with the checked input card image following it is written on the level 1 data file. The next input card is then read.

GABIN differs in one respect from DSTIN: tables of valid map sheet designations and card types are read first. After reading map sheet designations or card types on an input card these are converted into Volume and Page numbers by comparison with these tables. More Page types are allowed for than in the DST version.

3.2 Transfer Programs (DCARME, GCARME)

These programs merge the data from an existing level 2 file with new data from the level 1 file to form a new level 2 file.

The job steps are shown in Figure 7. By means of ATTACH statements the level 1 and level 2 files are made accessible. If, following a tape write step, the level 2 files had been cleared, a dummy is created by copying the level 1 files onto level 2. The FORTRAN program is executed, the old level 1 and level 2 files are cleared (PURGE) and the new files (temporarily named DCAT3, DFIL3) created by the merging are stored (CATALOG) as new level 2 files.

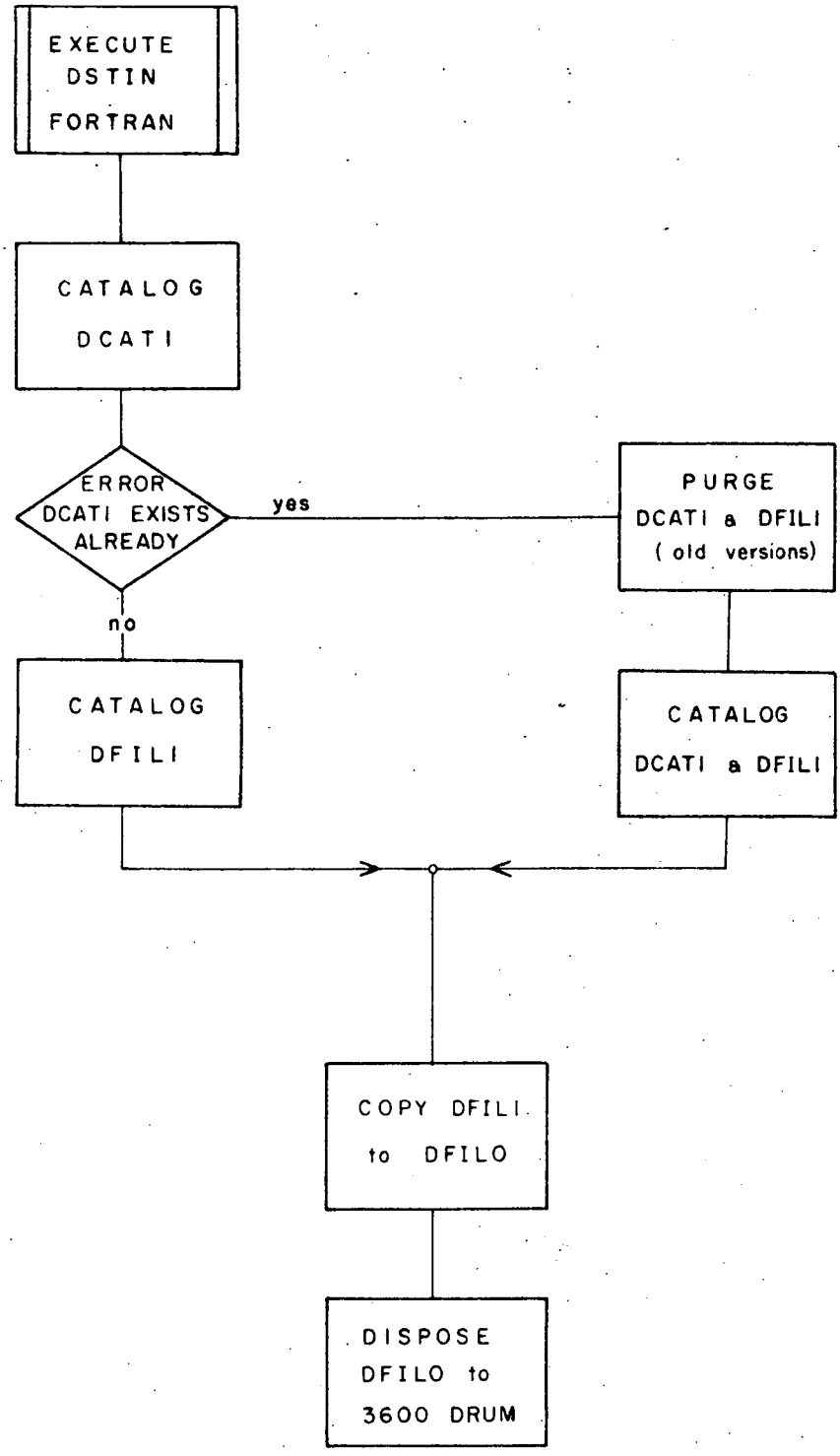
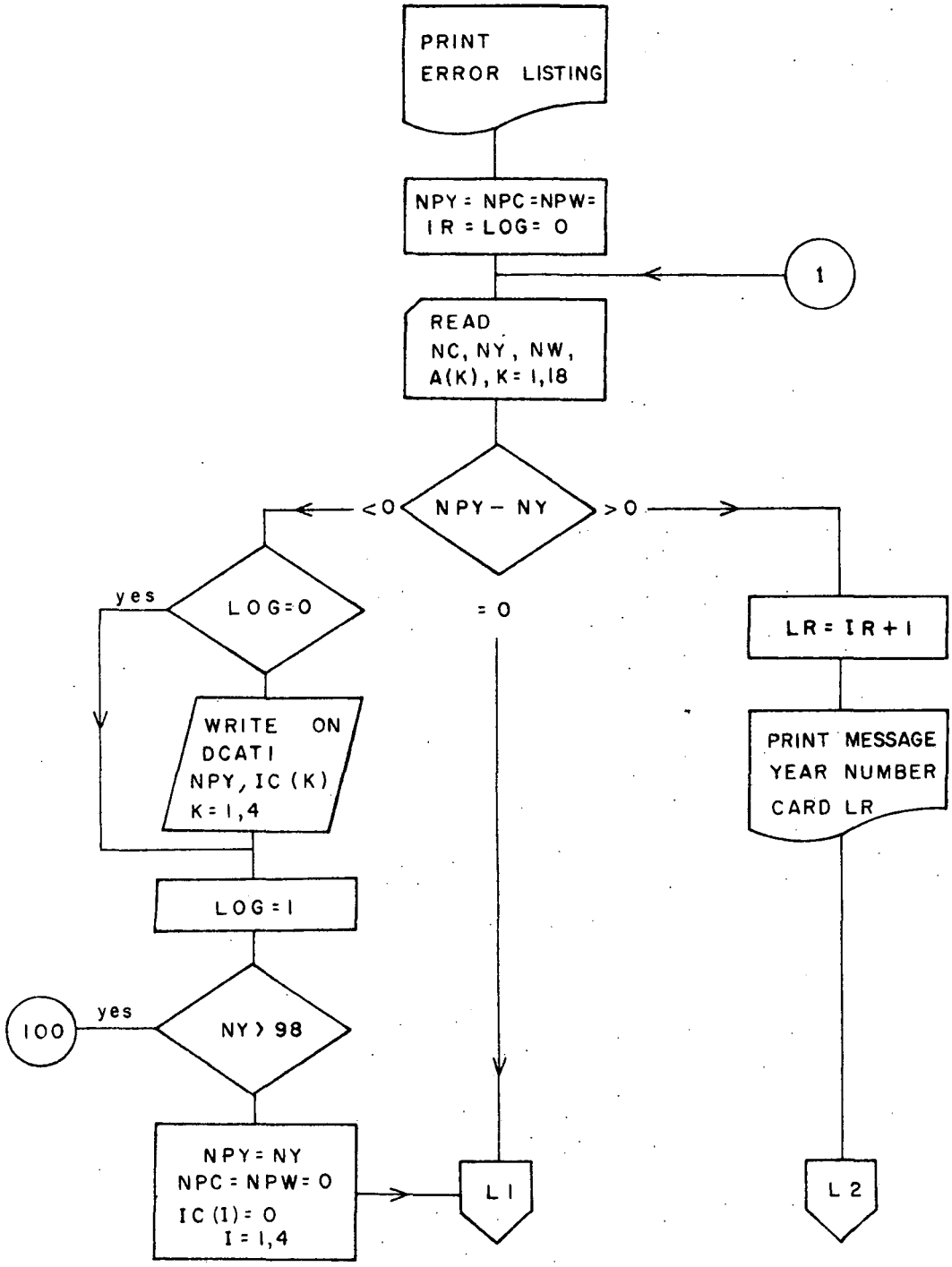


Fig.4 DSTIN flowchart job control statements.



continued Fig. 6

Fig. 5 DSTIN program flowchart. Part I

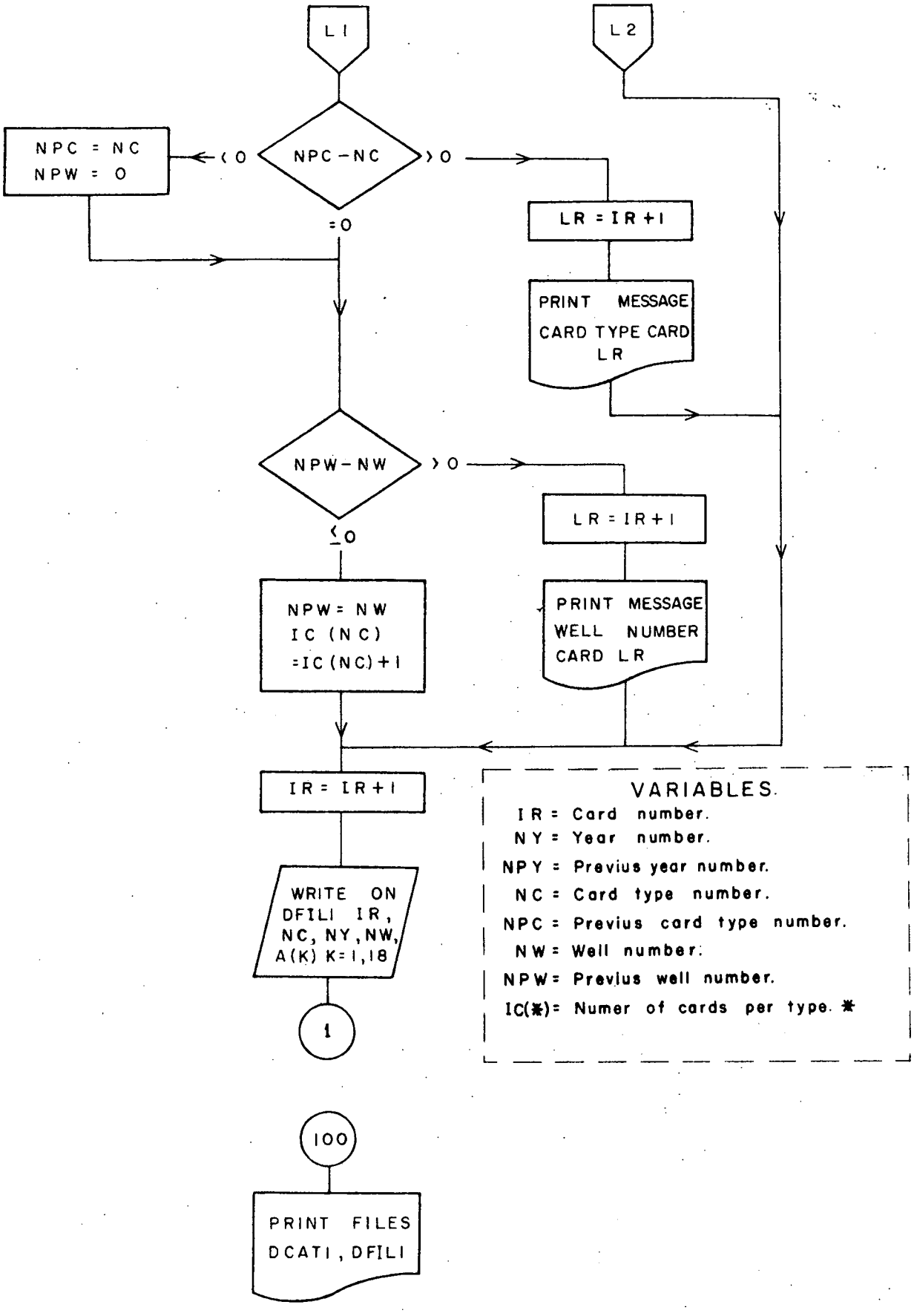


Fig. 6 DSTIN program flowchart Part 2

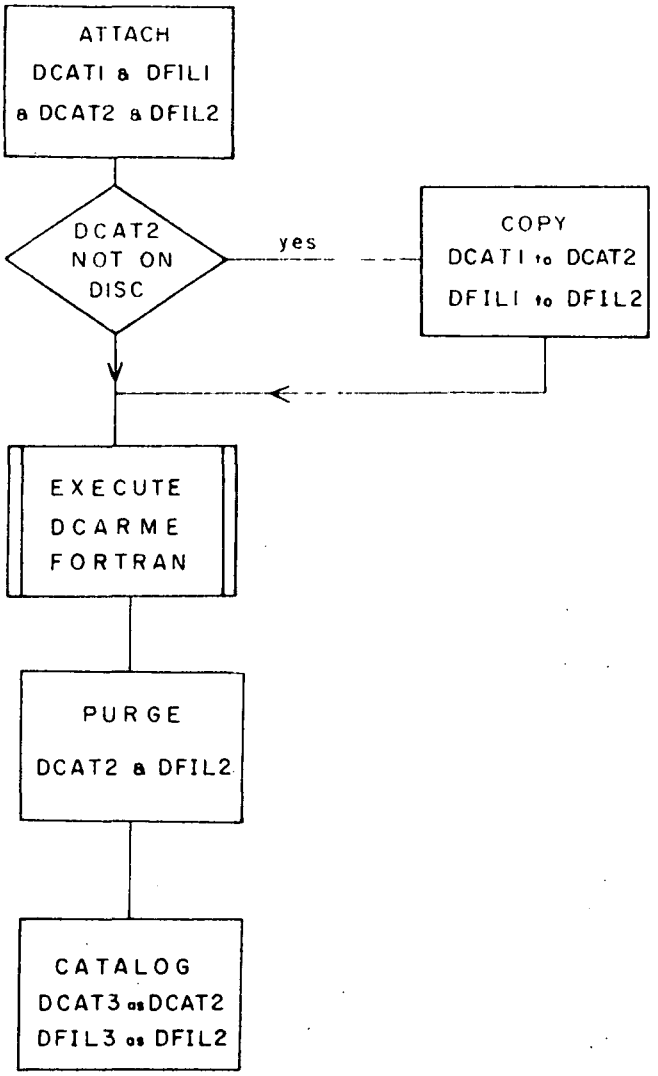


Fig. 7 DCARME flowchart job control statements.

Flow charts of program DCARME are presented in Figures 8 and 9. This program has as its only printout the catalogue of the new level 2 file. In a loop it reads a line of catalogue from DCAT1 (level 1) and a line of catalogue from DCAT2 (current level 2). If any of these files are at their end the Volume number (IYR (*)) for that file is set to 99 (999 for GAB). The Volume numbers on both catalogue lines are compared. If the Volume number in catalogue DCAT1 is less than the other (IYR(1) - IYR(2) 0) this volume has to be included first. The number of Lines to be copied (M) is calculated by adding together the numbers of Lines (IC (1,*)) for each Page. Those M Lines are copies from level 1 to the new level 2 data file and the complete level 1 catalogue line is included in the new level 2 catalogue file. A new line of catalogue is read from DCAT1, checked for EØF, and compared again to the current Volume number on DCAT2. Should the Volume number on DCAT1 be greater than the one on DCAT2 (IYR(1) - IYR(2) 0), corresponding action is taken but with the old level 2 files taking the place of the level 1 files.

If the Volume numbers in both catalogues are equal a comparison has to be made Page by Page - except when IYR(1) (and hence IYR(2) as well) is equal to 99, i.e. both files are at their end. Pages 1 are compared first. If the Page on the level 1 file is empty (IC (1,1) = 0) but not the one on level 2, then the level 2 Page is copied and ID(1), the number of Lines for Page 1 for the new catalogue Line, is set to the number of Lines on this Page IC(2,1), and vice versa if the Page on the level 2 file is empty but not the one on level 1. If both Pages contain Lines then the ones from level 1 will be copied and the ones from level 2 will be skipped. When all Pages have been compared a new line of catalogue is written for the new level 2 file.

3.3 Tape Write Programs (DMERGE, GMERGE)

These programs are very similar to the transfer programs described above. They merge the data from the current level 3 file on magnetic tape with the data from the level 2 file to create the new level 3 file on a new tape. The FORTRAN programs are identical to the ones described above except for the different file names.

The job steps are somewhat different as shown in Figure 10. The level 2 files DCAT2 and DFIL2 are accessed by ATTACH statements. The current tape is staged onto mass storage (DLEV2) and split by CØPYP into DCAT3 and DFIL3. The FORTRAN program is executed and the resulting new catalogue and data file (temporarily named DCAT4 and DFIL4) are copied and separated by an EØP (equivalent to EØF) mark onto a blocked file DLEV3, which then is staged out onto a new tape.

A copy of the level 2 files is preserved under the names DCAT2R and DFIL2R. Any old versions of the 'reserve' files are removed by PURGE commands if necessary. Finally the original level 2 files are cleared for new entries.

3.4 Retrieval and Update Programs (DRETRI, DTAPEX, GRETRI, GTAPEX)

The update programs are merely specialized versions of the retrieval programs combined with a version of the input program. The differences between the retrieve and update versions are quite elementary as apparent from comparison of the program print-out in Appendix B, and are not outlined any further here.

The job steps of the retrieval programs are shown in Figure 11. The current tape is staged in and split by copying into catalogue and data file. The FORTRAN program is executed. This program reads retrieval specifications, and copies the requested data onto the requested files. These files are then preserved by CATALOG statements.

To simplify the preparation of retrieval specifications a subroutine was devised, which accepts specifications in a compressed format and converts them into explicit form on file DCATE. Flow-charts for this subroutine INSPEC are shown in Figures 15 and 16. The compressed specifications may be in a variety of forms, which is explained further in Appendix A 4.1. INSPEC is a utility included for convenience and not strictly necessary. Hence, no further description is included here.

The flowcharts for program DRETRI are presented in Figures 12, 13, and 14. It calls subroutine INSPEC to obtain the explicit retrieval requests on file DCATE then reads two data cards. The first specifies the file numbers onto which individual Pages are to be copied. The second specifies whether EØF marks are to be inserted following each retrieved Page. The first retrieval specification is then read from file DCATE as Volume number and Page number, and then the first line of the catalogue DCAT3.

The Pages in the catalogue are then examined in a loop one by one. If a Page is empty (IC(*) = 0) the loop is advanced to the next Page until a non-empty Page is found. Then the Volume number IYE and Page number ICE in the retrieval specification is compared with the Volume number IYT and Page type number N in the catalogue. If any of the catalogue numbers is greater than the retrieval numbers (IYE-IYT 0 or ICE - N 0) a message is printed that the Volume and Page requested are not on tape, and a new

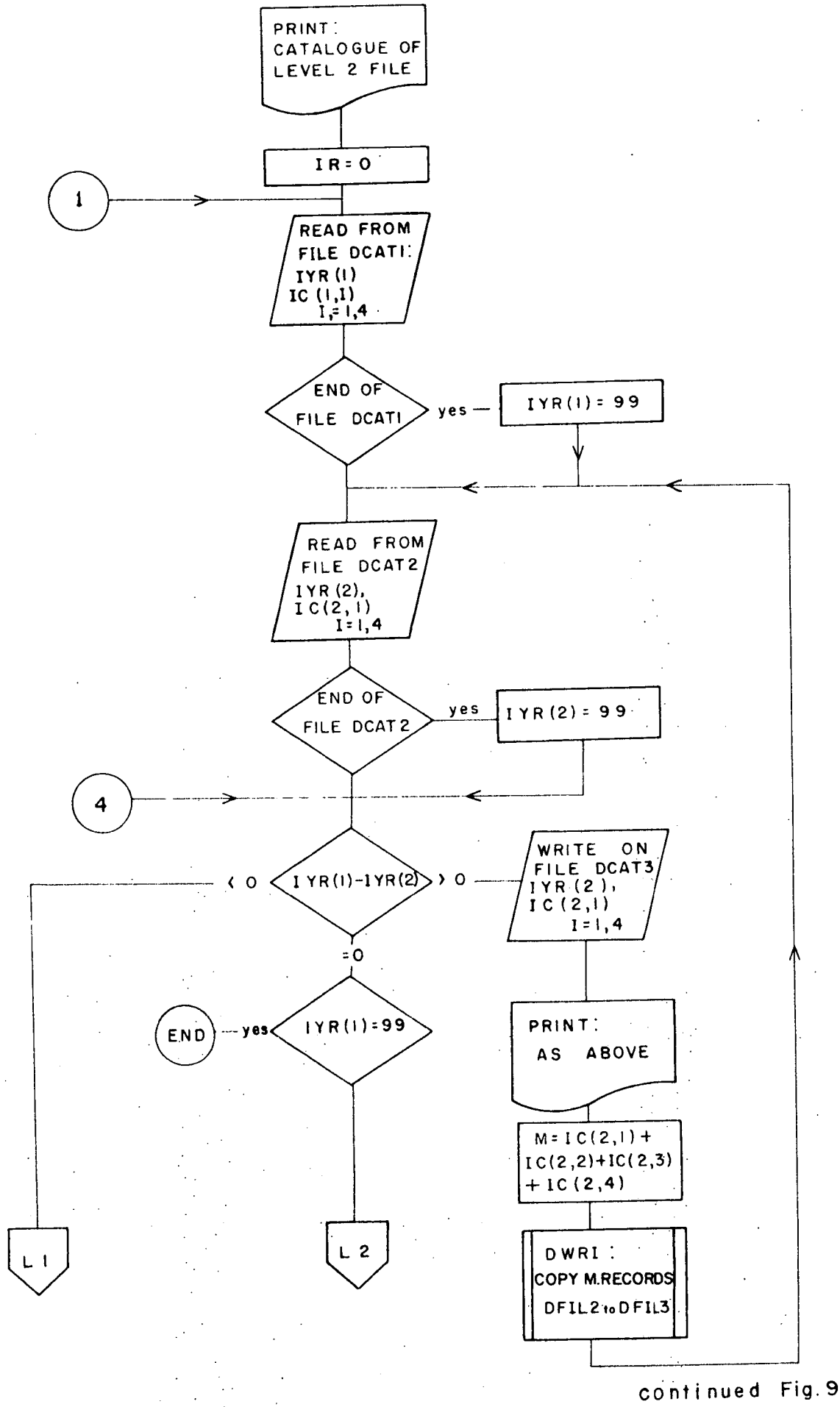


Fig.8 DCARME program flowchart. Part I

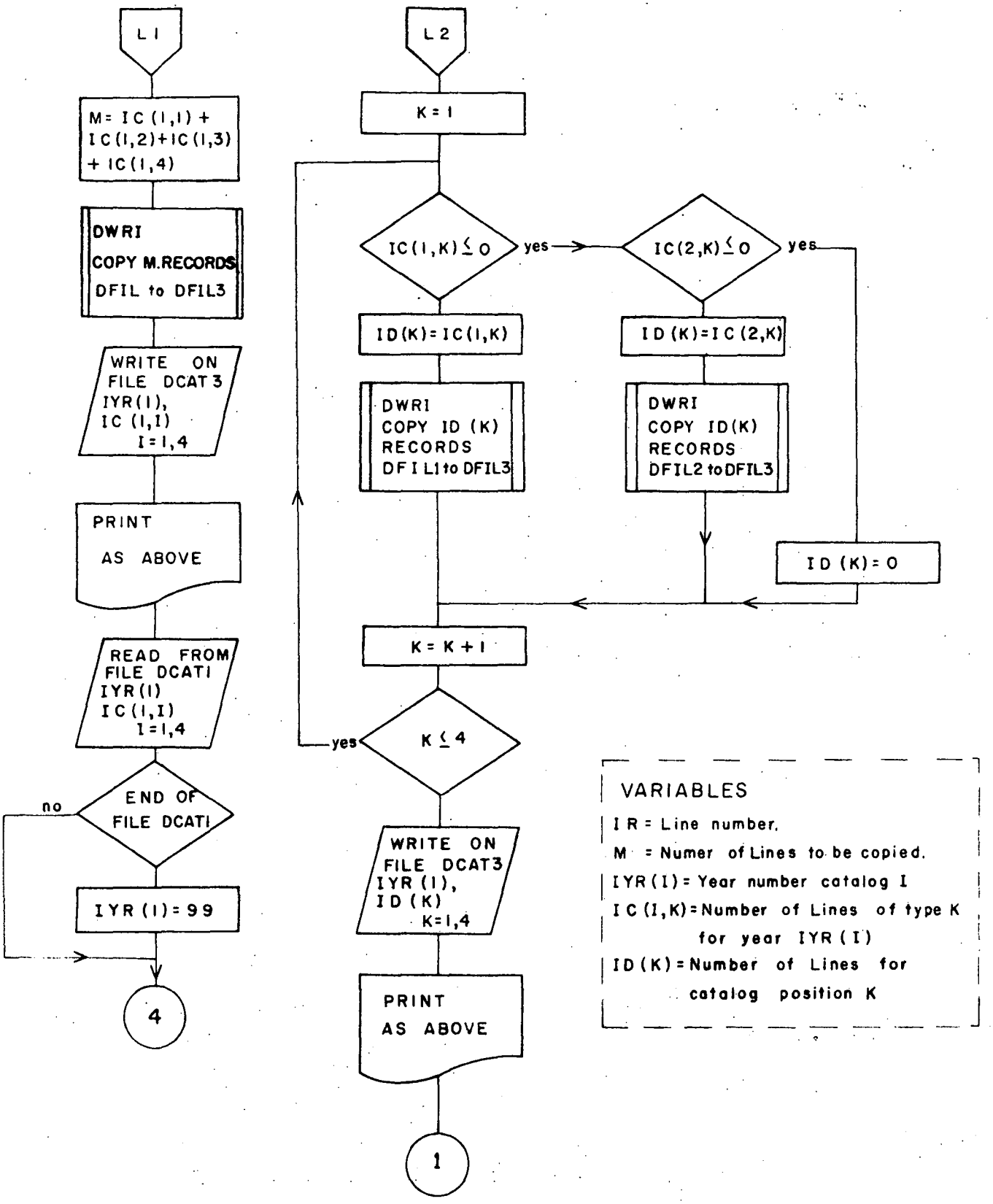


Fig. 9 DCARME program flowchart. Part 2

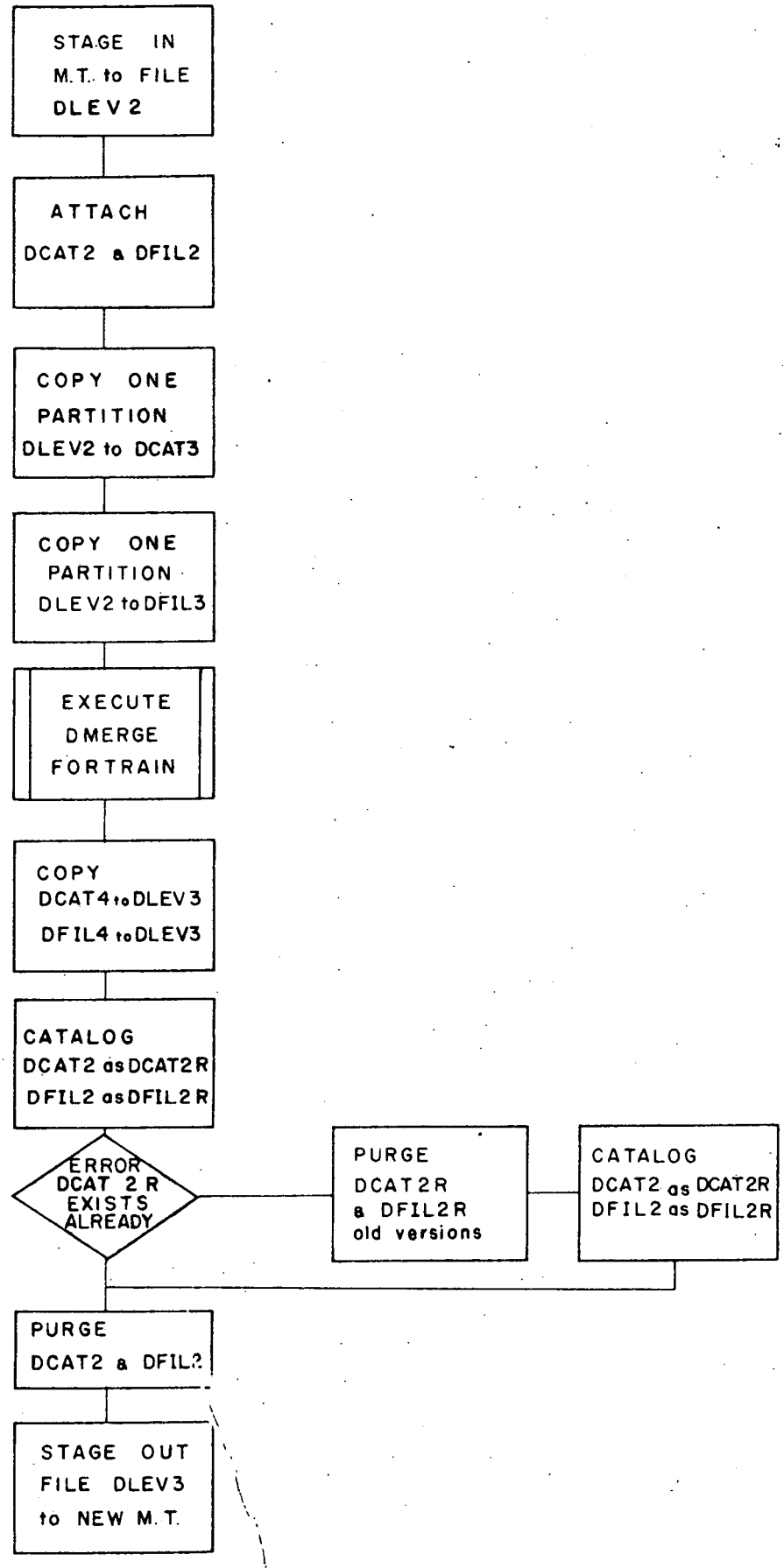


Fig. 10 DMERGE flowchart job control statements.

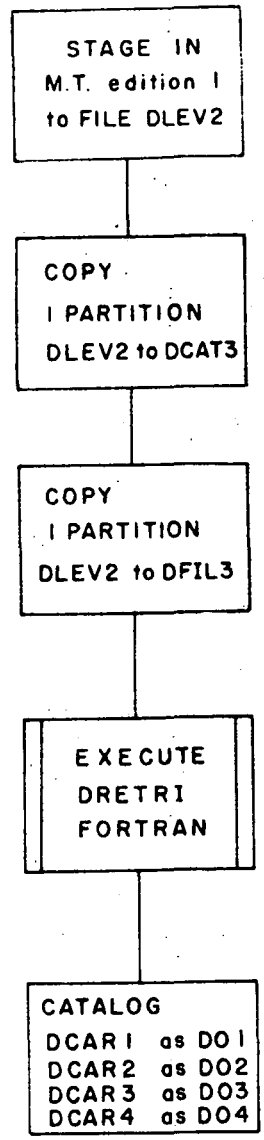
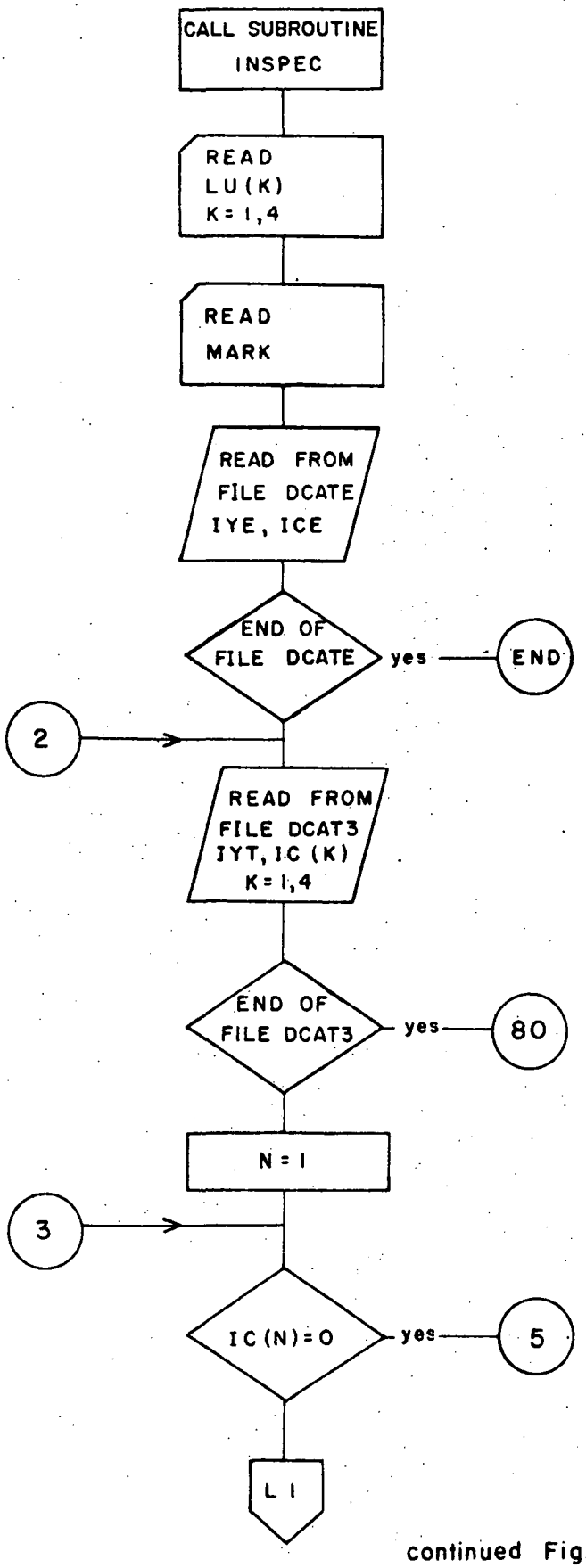
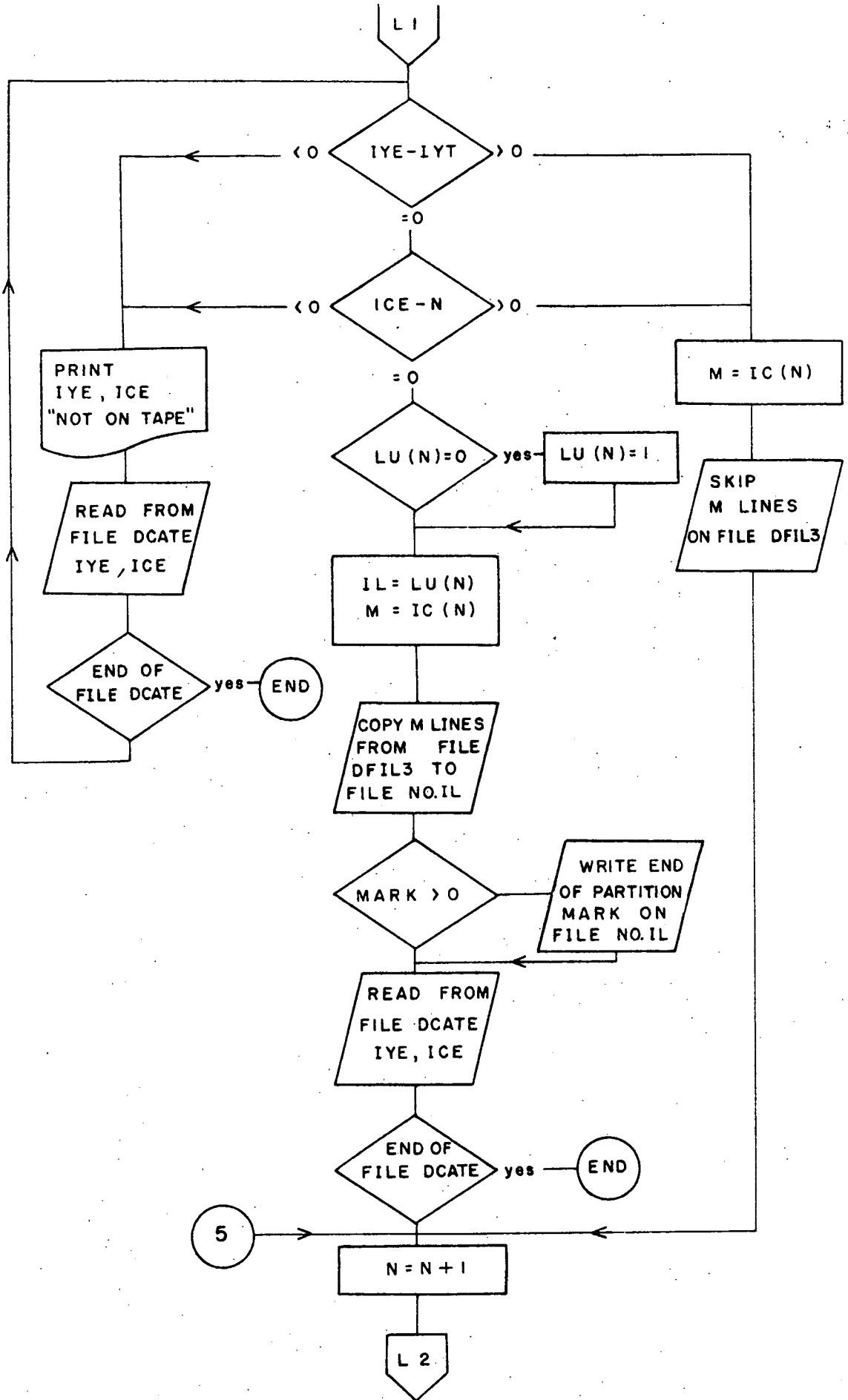


Fig.11 DRETRI flowchart job control statements.



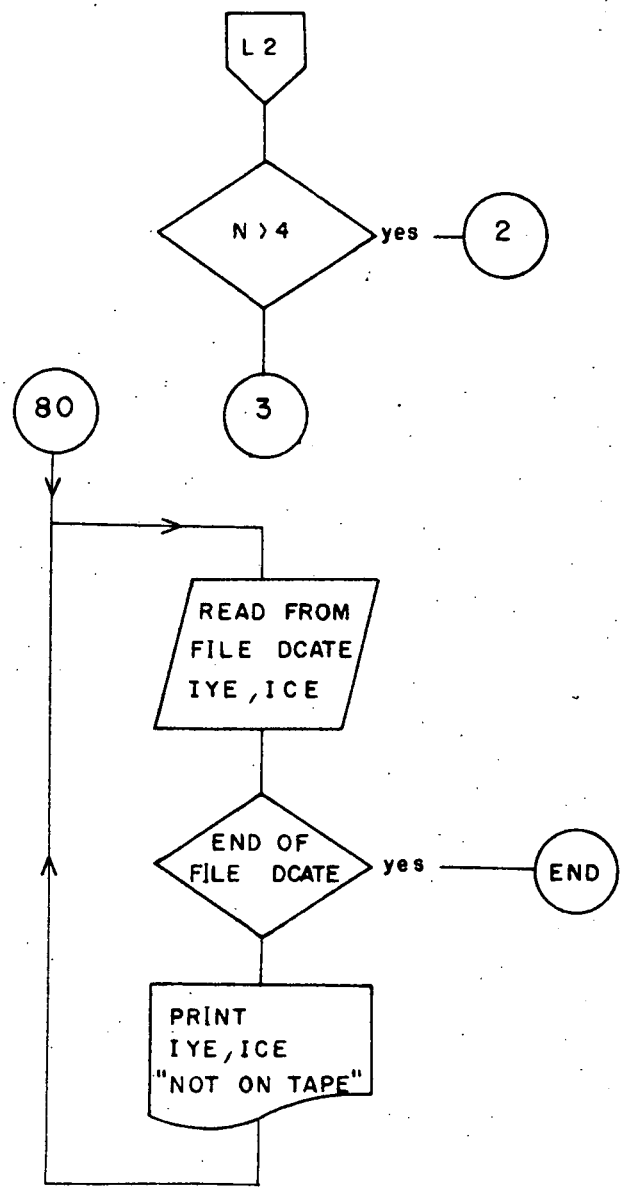
continued Fig.13

Fig. 12 DRETRI program flowchart Part 1



continued Fig.14

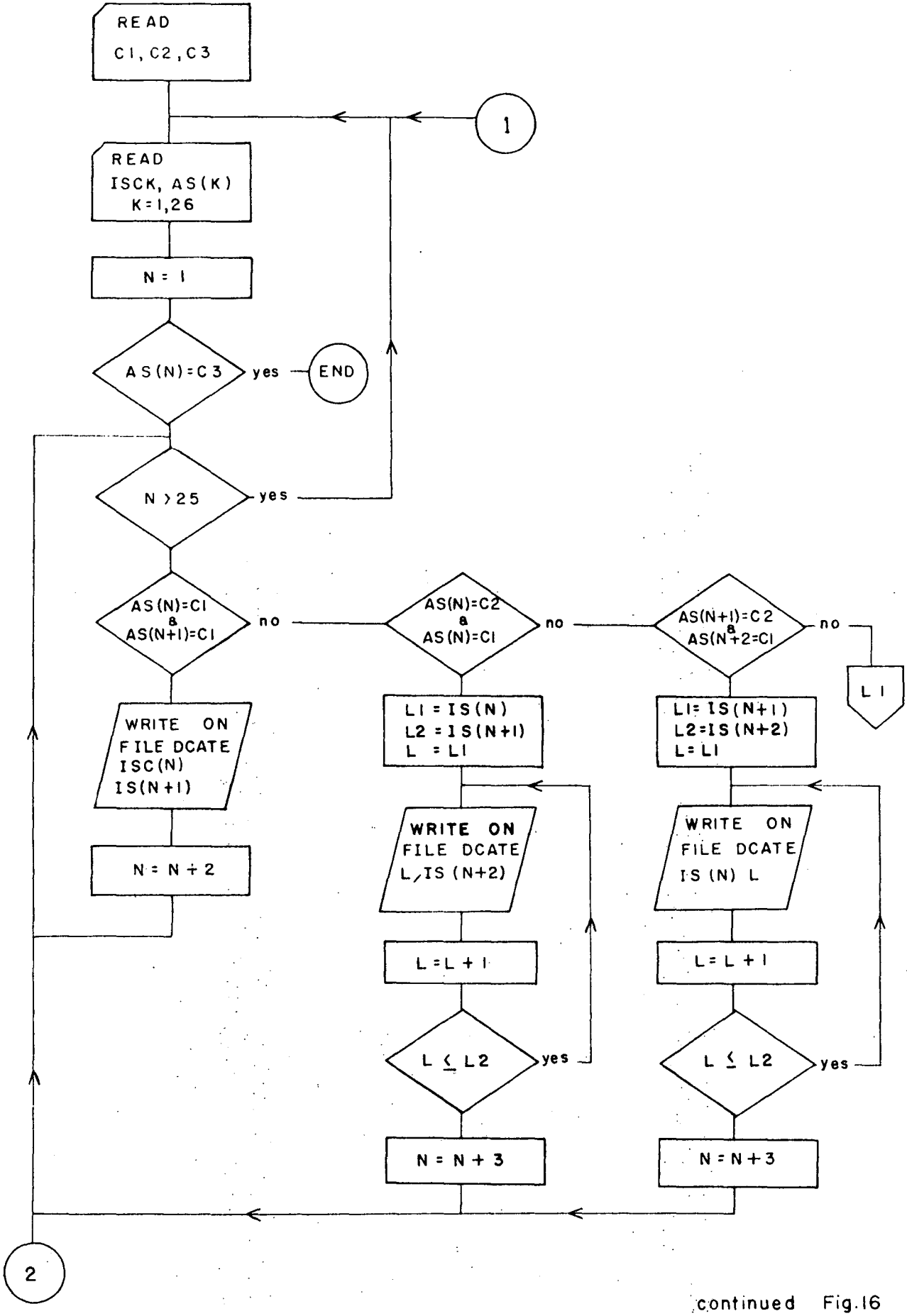
Fig. 13 DRETRI program flowchart. Part 2



VARIABLES

LU(*)	= File number for Page nos.*
MARK	= Indicator for EOF marks
IYE	= Year number (volume) to be extracted
ICE	= Card type (pages) to be extracted
IYT	= Year number on tape
IC(*)	= Number of Lines of Page *

Fig. 14 DRETRI program flowchart. Part 3



continued Fig.16

Fig.15 Subroutine INSPEC flowchart. Part 1

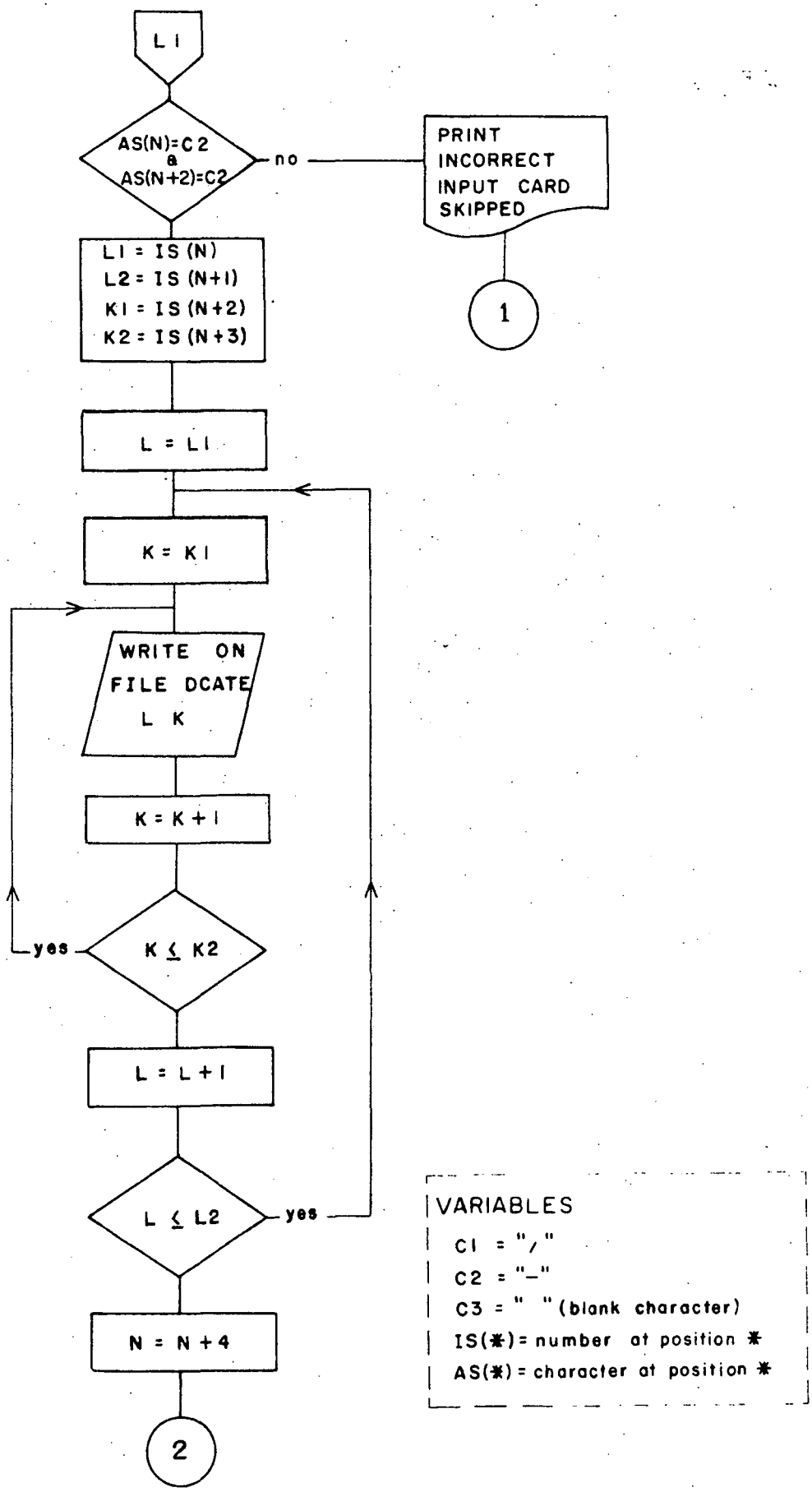


Fig.16 Subroutine INSPEC flowchart. Part 2

retrieval specification is read. If any of the retrieval specifications is less than the catalogue specifications the appropriate catalogue has not been reached yet and the appropriate number of Lines is skipped on the data file DFIL3.

When a match of retrieval specification and catalogue specification has been found the number of Lines given for the specified Page in the catalogue as IC(*) is copied onto the file number specified for that Page by LU(*). If LU was not specified, LU(*) = 1 is assumed. If EØF marks after each Page were requested by specifying MARK 0, such a mark is written on the same file. A new retrieval specification is read next.

After all Pages in a catalogue line have been examined a new line is read from DCAT3. If the end of the catalogue is reached before the end of retrieval specifications the remainder of the specifications is printed as not on tape.

4. PROCEDURES FOR USE OF THE SYSTEM

4.1 Notes on Terminal Operation

The complete system can be operated by submission of card decks for batch processing; however, some savings can be achieved by basing the operation of the system on a keyboard remote terminal system. In such a system, copies of data sets and programs are stored on disc or drum, modified by instructions, and submitted for processing from the terminal.

The language used to communicate with the computer through a terminal varies from one installation to another; so does the versatility of options offered. It is impossible therefore to devise a general system which fits all computer installations. However such a utility system is simple in design and should be used whenever possible.

One such system based on the CDC3600 and the CYBER76 computers has been designed for the CSIRO computer installation in Canberra. Detailed instructions on the use of this system are presented in Appendix A of this record. Instructions on the use of the complete system using card decks only are presented below.

4.2 Input and Editing of New Data

- Assemble the deck of data to be stored. Volumes may be incomplete (Pages missing) but each Page must be complete. Check for proper sorting of cards (see 2.1).
- Include data deck at the end of program deck DSTIN (GABIN for GAB system) just preceding the card with 999 punched in columns 1-3.
- Submit deck of program and data for batch processing and wait for results.
- Check print-out for errors in data. The print-out contains one or more pages headed'ERROR LISTING' indicating any cards out of sequence. This is followed by the data catalogue (DCATI) and a listing of all data cards now stored on DFILI each preceeded by a consecutive line number.
- If no errors were detected submit program deck DCARME to transfer the data to the next level and to free the data file DFILI for entry of new data; otherwise correct faulty data and resubmit as new data.
- When data are required on level 3 file (tape) continue with step 4.3.

4.3 Adding new Data to Permanent File

- Consult tape edition card to obtain serial number of current edition 1 tape.
- Change volume serial number (VSN=) on green STAGE (DLEV....) control card at front of card deck DMERGE to this number.
- Submit program DMERGE for processing.
- Printout will contain the catalogue of the new level 3 file and indicate the volume serial number of the new tape.
- Relabel on tape edition card old edition 1 - new edition 2, new tape - new edition 1, release old edition 2 tape.

4.4 Retrieval of Data for Processing

- Prepare a list of Volumes and Pages required, and code specifications as described in Appendix A 4.1 and punch on cards.

- Prepare one card to indicate file numbers on which Pages are to be stored (Appendix A4.2).

- Punch one card with any number in column 1 or 2 if EØF marks are required between Pages.

- Change tape volume serial number or green STAGE (DLEV2....) control card at front of program DRETRI to serial number of current edition 1 tape.

- At the end of the program deck DRETRI and preceding the last (EØI) card, include the following data cards:

  retrieval specifications

  one blank card

  card with file numbers (blank if file number 1 occurs for all Pages)

  card with number in column 1 or 2 on blank card.

- Submit completed deck for processing. Data will be available on the specified files. Data requested but not found on tape will be listed on print-out.

- A processing program can be submitted together with the retrieval program if dependency is specified (consult programmer). Otherwise submit processing program after print-out from DRETRI has been received.

4.5 Updating Data on Permanent File

- Correct card decks of Pages requireing updating.

- Resubmit these like new data through input sequence.

For this application, terminal operation of the system is particularly suitable (see Appendix A). Alternatively the following procedure may be followed:

- Prepare STAGE (DLEV.....) card of program DRETRI and retrieval specifications as described in 4.4.

- Retrieve affected Pages as described.

- Alter data under program control (e.g. conversion of certain variables into different units) and machine-punch new card deck of data.

- If alternation is permanent, resubmit like new data.

5. REFERENCES

UNGEMACH, P., & HABERMEHL, M.A. - Great Artesian Basin Groundwater Project automatic data processing storage and retrieval system. Bur. Miner. Resour. Aust. Rec. 1973/25.

KREBS, G. - Data check programs for the Great Artesian Basin ADP System. Bur. Miner. Resour. Aust. Rec. 1973/203.

APPENDIX A

TERMINAL OPERATION FOR GAB-DST DATA STORAGE AND RETRIEVAL SYSTEMS

INSTRUCTIONS FOR USE

Contents

TERMINAL OPERATION FOR GAB-DST STORAGE AND RETRIEVAL SYSTEMS

A1. INTRODUCTION

The terminal operation described here is specific to the DAD system implemented on the CDC 3600 by the Division of Computing Research of the CSIRO, and cannot simply be transferred to other installations. The 3600 at present is linked to the new CYBER 76 computer allowing some interactive operation. This link is not permanent and the life of this system is limited. However the use of the terminal system results in much faster processing than physical submission of card decks and it is expected that the input, editing, and updating phase for the GAB and DST data will be completed before the 3600 is taken out of service.

The terminal system can be used to submit programs and to edit data. Submitting programs by terminal is very simple and does not require any knowledge of the DAD system apart from the instructions presented below. Editing of data through the terminal requires knowledge of some of the TED commands (CSIRO TED Manual). These commands are simple and can be learned quickly.

A2. OPERATION OF TERMINAL

A2.1 Keyboard Use

The keyboard on the STC Teletype terminal resembles an ordinary typewriter keyboard with some special keys added. Keys of particular importance are:

	Abbreviated
Shift key	(S)
Control character key	(C)
Line feed key	(L)
Carriage return key	(R)

Often these special keys have to be depressed simultaneously with other keys. Simultaneous depression of, for example, the control character key with the ordinary P key is represented here as (C)P, and accordingly for other combinations, e.g. (S)M, (S)?, (S)K, ... Depressing the (L) key transmits a completed line to the computer; depressing the (R) key cancels a line which has been typed but not yet transmitted.

The sequence of typing a line on the keyboard followed by depression of (L) is referred to as 'typing in' of that line.

A2.2 Logging in and out

To activate the link between the terminal and the computer and to get the system ready to accept instructions from the terminal is referred to as 'logging in'. 'Logging out' means to terminate the session and to disconnect the terminal.

- To log in, first turn the knob on the right-hand side of the terminal to the 'Line' position.
- Type - (C)P followed by (S)? ; this will produce a reply whether the system is operative or not.
- Type in - *LG,CCDMR*WC,TED ; the system (if operative) will reply with the time, date, TED 3.0, and finally a colon on a new line.
- Type in - (S)K/DINBØX/ for the DST, (S)K/GINBØX/ for the GAB system. (S)K is printed as . The system is now ready to accept instruction and the special commands listed below.
- To log out after completion of session type in - Q

A2.3 Abnormal Conditions

Erratic response of system. Minor delays in response to a command are common. If, however, delays are excessive, e.g. more than 30 seconds, the system is probably overloaded or not functioning properly; therefore, return control to the keyboard by typing (C)P and A. This should result in a colon being printed on a new line. Then log out by typing in Q. If this is unsuccessful the sequence (C)P and T can be used to terminate the session.

Wrong character(s) typed. As long as the characters have not yet been transmitted by depressing the (L) key there are two methods of correction. The sequence (C)B repeated times will remove the last characters, or (R) will cancel the current line and return the carriage for re-entry of the corrected line.

Message DOCUMENT NOT ON DR. This means that a document (data set or program) referenced by a command is not available. Check whether all standard data sets and programs are still available by typing in - 1Ø. If any of these data sets are missing they have to be restored by submitting the appropriate card deck. The only other data set referenced by any of the commands is created by the input programs. If this data set is unavailable the appropriate input program has to be run again.

A3. INPUT SEQUENCE

A3.1 First Input of Data

- Check that deck of data cards is complete (all cards of a particular card type and map sheet or year number must be present) and correctly sorted.
- Include data deck at the end of program deck (GABIN for GAB, DSTIN for DST data) just preceding the card with 999 in columns 1-3.
- Submit deck of program and data for processing and wait for print-out.
- Log in on terminal (see A2.2).
- Type in 10$ to preserve data set.
- Log out.
- Check print-out for errors in data. The print-out contains one or more pages headed 'ERROR LISTING', indicating cards out of sequence. This is followed by the data catalogue and a listing of all data cards each preceded by a line number.
- If cards were out of sequence correct data deck and repeat A3.1.
- If punch errors were detected continue with A3.2 or repeat A3.1. with corrected data.
- If data are correct continue with A3.3.

A3.2 Editing of Data

- Log in on terminal.
- Type in - C/N36/ to obtain working copy of data file.
- Use TED commands to change data file.
- Type in - M/N36/ to store corrected copy.
- Type in - 11$ to merge data with program.
- Type in - 12$ to submit job for processing.
- Log out.
- Continue as in A3.1 when print-out arrives.

A3.3 Transfer of Data to Level 2 Storage

- Log in on terminal.
- Type in 13$; this submits completed data with program for processing.
- A print-out will be received listing the catalogue of the new combined level 2 file.
- After this print-out has been received new data may be entered through step A3.1.
- If the data entered in level 2 are required for processing, or an updating of level 3 (magnetic tape) is required for other reasons, proceed with step A3.4. No further action otherwise.

A3.4 Transfer of Data to Level 3 (New Tape)

- Log in on terminal.
- Type in 14$.
- Type in I/XXXX/ ; where XXXX is the four digit serial number of the current-edition 1 tape (from tape edition card).
- Type in 15$; this submits job to CYBER 76 for processing.
- Print-out will list new catalogue of combined data on new tape-edition 0 .
- Tape-save record card will be forwarded by CSIRO containing tape serial number of new tape.
- If tape writing is not successful, consult programmer.
- If tape writing is successful, release old-edition 2 tape; relabel on tape-edition card, old-edition 1 tape as new-edition 2 (back-up tape), and edition 0 tape as new-edition 1 (current tape).

A4. RETRIEVAL SEQUENCE

A4.1 Formats for Retrieval Specifications

Retrieval is by Volume and Page. For the DST system the definitions are Volume = year number, Page number = card type; and for the GAB system Volume number = map sheet number, Page number = card type number. A special table has been prepared for the GAB system, listing the assigned volume and page numbers. In the GAB system these numbers are three digits; in the DST system, two digits. The following symbols are used in the table below

X1, X2, X3 Volume numbers two or three digits

Z1, Z2, Z3 Page numbers two or three digits

Type	Specification	Volumes retrieved	Pages retrieved	Valid for
1	X1,Z1	X1	Z1	GAB, DST
2	X1-X2,Z1	X1 - X2 incl	Z1	GAB, DST
3	X1,Z1-Z2	X1	Z1-Z2 incl.	GAB, DST
4	X1-X2,Z1-Z2	X1 - X2 incl.	Z1-Z2 incl.	GAB, DST
5	X1-X2,Z1.Z2.Z3	X1 - X2 incl.	Z1 and Z2,Z3	GAB only

X1 and X2 may be any valid Volume number as long as X2 is greater than X1 the same applies to the Page numbers. Specification types may be mixed. Example: to retrieve Volume 63, Page 5; Volumes 65-69, Page 11; and Volumes 112-114, Pages 6, 11, and 13, the specification is (GAB system):

063,005,065-069,011,112-114,006.011.013

Specifications may extend over one or more lines and must start in column 1 of each line.

Example for DST:

61,04,62,01-03,65-68,04,
70-73,01-04.

A4.2 Selection of Output Unit and Output Grouping

Normally all retrieved data are stored on a permanent disc file D01 or G01 in a continuous block. However the output may be spread and grouped if required. Up to thirteen different storage files are available on the GAB, and up to four on the DST system. Each card type may be stored on a different file by specifying a file number for individual card types in the range 01 to 04 for the DST,

or 01 to 13 for the GAB system. Specification is by one line of input; where the columns 1 and 2 specify the file number for pages 1 (card type number 1), and columns 3 and 4 specify the file number for pages 2; columns 39 and 40 specify the file number for pages 20. If any columns are left blank, file 01 is assumed.

Example

0413 060711

pages 1 on file 04 (GO4 for GAB, DO4 for DST)

pages 2 on file 13

pages 3 on file 01

pages 4 on file 06

pages 5 on file 07

pages 6 on file 11

pages 7 to pages 13 on file 01

A blank line of course has the effect of assigning file 01 to all pages.

Optionally Pages may be separated on their output files by End of File Marks (EØF). This is specified by an input card with a positive number in columns 1 and 2. If zero or blanks are specified on this card no EØF marks will be inserted. Since there is only one Page of a particular Page number to each Volume, specification of EØF marks has the effect of separating Volumes on any files containing only one Page type.

A4.3 Operation of Retrieval Sequence

- Log in on terminal.
- Type in 20$; this effects clearing of the extraction files.
- Type in 21$.
- Type in I/XXXX/; where XXXX is the tape serial number of the current-edition 1 tape.
- Type in 22$.
- Type in Z/ab,cd, ./J ; where ab,cd, , is the retrieval specifications (see 4.1). Repeat if more than one line of specifications.
- Type in Z/ /J; some blank characters between / /.

. Type in Z/abcdef /j ; where abcdef are the file number specifications (see 4.2).

. Type in Z/xy/J ; where xy is a number greater than zero or blank depending on whether EØF marks are to be inserted (see 4.2).

. Type in 23$, to submit program for processing.

. Print-out will contain appropriate messages if map-sheet page combinations requested for retrieval were not on tape. All other requested data will be stored on the specified files.

A5 UPDATING SEQUENCE

A5.1 Procedure

The updating sequence is a combination of retrieval and input. In the simplest case, where the data set is to be expanded by insertion of additional cards, retrieval is not required. The additional data cards are inserted into the original card deck and all pages affected by insertion are resubmitted through the normal input sequence and will replace the old versions on tape. Where individual characters or cards are to be changed by editing the procedure described below may be followed instead.

A5.2 Retrieval

. Log in on terminal.

. Type in 30$ to clear extraction files.

. Type in 31$.

. Type in I/XXXX/; where XXXX is the tape serial number of the current-edition 1 tape.

. Type in 32$.

. Type in Z/ab,cd, ,/j ; where ab,cd , is the retrieval specifications as defined in 4.1.

. Type in Z/ /J,; repeat this line once more.

. Type in 33$ to submit for processing.

. Print-out will be received as if retrieval data had been submitted as card deck through input program DSTIN or GABIN.

A5.3 Editing and Updating

Continue with Input Sequence starting with instruction 3 ('Wait for printout') of A3.1.

A6. PROGRAM MAINTENANCE

Copies of all programs and of special instruction sequences are stored on the document region of the 3600. Their presence there is necessary for the terminal operation of the system. To retain these documents on the drum their retention period has to be extended regularly. This can be done by executing 'LOCATE' commands for all documents. Such commands are prestored as a document. The procedure for extending the retention period is:

- Log in on terminal.
- Type in 1Ø; the system will reply by printing the nine LOCATE commands for the various documents.
- If any of the documents is not located, which is indicated by DOCUMENT NOT ON DR, consult a programmer to restore the missing document(s) (see also A2.3).
- Log out.

This operation should be carried out once a week preferably, but never less than once a fortnight. Otherwise the documents will be flushed from the drum.

APPENDIX B

SAMPLE PRINT-OUTS OF PROGRAMS AND DATA

```
** CYBER76 SCOPE 2.0 LVL 65  08/11/73  13.20.36   14/11/73      73313

CSIRO DAD3.30/SCOPE2.0 CSIRONET SYSTEM MOD65D.                                                    B 1

HH.MM.SS CPU SECOND ORIGIN
         13.56.45 STN,      DIOS  VERSION 24/10/73.
         13.56.45 STN,         228 RECORDS READ.
13.56.46 00000.006 JOB,     -DSTIN(T30,P1000)                                          DSTIN  Versions 1 and 2
13.56.46 00000.007 JOB,     -ACCOUNT(CCDMR*WC)
13.56.46 00000.007 LOD,     -FTN.
13.56.47 00000.174 USR,           .159 CP SECONDS COMPILATION TIME
13.56.47 00000.175 LOD,     -LGO.
13.56.47 00000.208 SYS,       LD610 - FLS REQUIRED TO LOAD = 0006420 DU.COG
13.56.47 00000.209 SYS,       LD603 - EXECUTION INITIATED OS.EXP
13.56.47 00000.209 USR,      FORTRAN LIBRARY 65          24/08/73
13.56.47 00000.440 USR,         END  DSTIN
13.56.47 00000.442 USR,           .232 CP SECONDS EXECUTION TIME
13.56.47 00000.443 JOB,     -CATALOG(DCAT1,DCAT1,ID=CCDMR*WC,RP=15)
13.56.47 00000.444 SYS,       PF030 - PF NAME ALREADY IN SYSTEM
13.56.48 00000.477 SYS,       SC031 - JOB ABORTED
13.56.48 00000.479 JOB,     -EXIT.
13.56.48 00000.480 JOB,     -ATTACH(ERCAT,DCAT1,ID=CCDMR*WC)
13.56.48 00000.482 SYS,       PF673 - CYCLE     1 ATTACHED
13.56.48 00000.483 JOB,     -ATTACH(ERFIL,DFIL1,ID=CCDMR*WC)
13.56.48 00000.485 SYS,       PF673 - CYCLE     1 ATTACHED
13.56.48 00000.485 JOB,     -PURGE(ERCAT)
13.56.48 00000.486 SYS,       PF256 - FILE SUCCESSFULLY PURGED
13.56.48 00000.487 JOB,     -PURGE(ERFIL)
13.56.48 00000.488 SYS,       PF256 - FILE SUCCESSFULLY PURGED
13.56.48 00000.489 LOD,     -UNLOAD(ERCAT,ERFIL)
13.56.49 00000.492 JOB,     -CATALOG(DCAT1,DCAT1,ID=CCDMR*WC,RP=15)
13.56.49 00000.494 SYS,       PF060 - CYCLE     1 CATALOGED
13.56.49 00000.495 JOB,     -CATALOG(DFIL1,DFIL1,ID=CCDMR*WC,RP=15)
13.56.49 00000.497 SYS,       PF060 - CYCLE     1 CATALOGED
13.56.49 00000.497 LOD,     -REWIND(DFIL1)
13.56.49 00000.500 LOD,     -COPY(DFIL1,DFIL0)
13.56.49 00000.524 USR,       UT031 - EOI ENCOUNTERED
13.56.49 00000.525 USR,       UT034 -     EOR = 141          EOS =         0    EOF = 2
13.56.49 00000.525 LOD,     -REWIND(DFIL0)
13.56.49 00000.528 JOB,     -DISPOSE(DFIL0,*DR*CPR)
13.56.49 00000.532 SYS,       RM770 - MAXIMUM ACTIVE FILES                  5
13.56.49 00000.532 SYS,       RM771 - OPEN/CLOSE CALLS                     31
13.56.49 00000.532 SYS,       RM772 - DATA TRANSFER CALLS                1,574
13.56.49 00000.532 SYS,       RM773 - CONTROL/POSITIONING CALLS            39
13.56.49 00000.532 SYS,       RM774 - BM DATA TRANSFER CALLS              258
13.56.49 00000.533 SYS,       RM775 - BM CONTROL/POSITIONING CALLS         71
13.56.49 00000.533 SYS,       RM776 - QUEUE MANAGER CALLS                  57
13.56.49 00000.533 SYS,       RM777 - RECALL CALLS                         80
13.56.49 00000.534 SYS,       SCM                  4.588  KWS
13.56.49 00000.534 SYS,       I/O                  0.013  MS
13.56.49 00000.534 SYS,       RMS                  0.006  MWS
13.56.49 00000.534 SYS,       USER                 0.256  SEC
13.56.49 00000.535 SYS,       JOB                  0.536  SEC
13.56.49 00000.535 SYS,       SC050 - 000000 SC/LC SWAPS
13.56.49 00000.535 SYS,       PR;                     1
13.56.49 00000.535 SYS,       CRC                  1.075  SEC
13.56.49 00000.535 SYS,       TOT                  1.924  SEC
```

ERROR HANDLING

```
** CYBER76 SCOPE 2.0 L/L 65 - 08/11/73  13.21.36   13/11/73      73317

CSIRO BA03.3J/SCOPE2.0 CSIRONET SYSTEM MOD69D.

HH,MM,SS CPU SECOND ORIGIN
         14.06.30 STN,      DIOS  VERSION 24/11/73.
         14.06.30 STN,         83 RECORDS READ.
14,13,51 00010.015 JOB,    -DSTIN3(T20,P100),B4201)                                    B 2
14,13,51 00010.016 JOB,    -ACCOUNT(CCDMR*C)
14,13,51 00010.016 JOB,    -ATTACH(DCAR11,DCAT4,ID=CCDMR*C)
14,13,51 00010.019 SYS,       PF673 - CYCLE     1 ATTACHED
14,13,51 00010.019 LOD,    -FTN,                                                       DSTIN Version 3
14,13,54 00010.177 USR,          .159 CP SECONDS COMPILATION TIME
14,13,54 00010.178 LOD,    -LGO.
14,13,56 00010.211 SYS,       LD610 - FLS REQUIRED TO LOAD = 0006420 CU,COG
14,13,56 00010.212 SYS,       LD603 - EXECUTION INITIATED 25,EXP
14,13,56 00010.212 USR,     FORTRAN LIBRARY 83            24/03/73
14,13,56 00010.414 USR,          END  DSTIN
14,13,56 00010.445 USR,          .232 CP SECONDS EXECUTION TIME
14,13,56 00010.446 JOB,    -PURGE(DCAR11)
14,13,56 00010.447 SYS,       PF256 - FILE SUCCESSFULLY PURGED
14,13,56 00010.448 JOB,    -CATALOG(DCAT4,DCAT4,ID=CCDMR*C,RP=15)
14,13,57 00010.430 SYS,       PF060 - CYCLE     1 CATALOGED
14,13,57 00010.431 JOB,    -CATALOG(DFIL1,DFIL1,ID=CCDMR*C,RP=15)
14,13,57 00010.433 SYS,       PF060 - CYCLE     1 CATALOGED
14,13,57 00010.434 LOD,    -REWIND(DFIL1)
14,13,57 00010.436 LOD,    -COPY(DFIL1,DFIL)
14,13,57 00010.431 USR,       UT031 - EOI ENCOUNTERED
14,13,57 00010.431 USR,       UT034 -      EOR = 141          EOS =          0    EOP = 2
14,13,57 00010.432 LOD,    -REWIND(DFIL1)
14,13,58 00010.435 JOB,    -DISPOSE(DFIL1,*DR=CPR)
14,13,58 00010.435 JOB,    -EXIT.
14,13,58 00010.439 SYS,       RM770 - MAXIMUM ACTIVE FILES                            6
14,13,58 00010.439 SYS,       RM771 - OPEN/CLOSE CALLS                               31
14,13,58 00010.439 SYS,       RM772 - DATA TRANSFER CALLS                         1,679
14,13,58 00010.440 SYS,       RM773 - CONTROL/POSITIONING CALLS                      38
14,13,58 00010.440 SYS,       RM774 - BM DATA TRANSFER CALLS                        242
14,13,58 00010.440 SYS,       RM775 - BM CONTROL/POSITIONING CALLS                   71
14,13,58 00010.440 SYS,       RM776 - QUEUE MANAGER CALLS                            55
14,13,58 00010.440 SYS,       RM777 - RECALL CALLS                                   71
14,13,58 00010.471 SYS,       SCM                  4.358  K.S
14,13,58 00010.471 SYS,       I/O                  1.312  MS
14,13,58 00010.471 SYS,       RMS                  0.005  K.S
14,13,58 00010.472 SYS,       USER                 1.259  SEC
14,13,58 00010.472 SYS,       JOB                  0.494  SEC
14,13,58 00010.472 SYS,       SC050 - 000000 SC/LC SWAPS
14,13,58 00010.472 SYS,       PRI                     1
14,13,58 00010.472 SYS,       URC                  1.519  SEC
14,13,58 00010.473 SYS,       TOT                  1.319  SEC
```

```
      PROGRAM DSTIN(INPUT,OUTPUT,DCART1,DFIL1,DCAR1,TAPE1=DCAR1,TAPE4=
     1OUTPUT,TAPE2=DFIL1,TAPE3=DCART1)
      DIMENSION IC(4),A(20)
      WRITE(4,800)
  800 FORMAT(14H1ERROR LISTING)
      NPY=NPC=IR=NPW=LOG=0                                    B 3
    1 READ(1,90) (NC,NY,NW,(A(K),K=1,18))
   90 FORMAT(I1,I2,I5,18A4)
      IF(NPY-NY) 3,4,50
   10 3 IF(LOG.EQ.0) GO TO 5                                  DSTIN Versions 1 and 3
      WRITE(3,900) (NPY,(IC(K),K=1,4))
  900 FORMAT(I2,2X,4I6)
    5 LOG=1
      IF(NY.GT.75) GO TO 100
   15 NPY=NY
      NPC=NPW=0
      DO 6 I=1,4
      IC(I)=0
    6 CONTINUE
   20 4 IF(NPC-NC) 7,9,60
    7 NPC=NC
      NPW=0
    9 IF(NPW-NW) 8,8,70
    8 NPW=NW
   25 IC(NC)=IC(NC)+1
   10 IR=IR+1
      WRITE(2,901) (IR,NC,NY,NW,(A(K),K=1,18))
  901 FORMAT(I6,4X,I1,I2,I5,18A4)
      GO TO 1
   50 LR=IR+1
      WRITE(4,902) LR
  902 FORMAT(1H0,*YEAR NUMBER CARD*,I6)
      GO TO 10
   60 LR=IR+1
      WRITE(4,903) LR
  903 FORMAT(1H0,*CARD TYPE CARD*,I6)
      GO TO 10
   70 LR=IR+1
      WRITE(4,904) LR
  904 FORMAT(1H0,*WELL NUMBER CARD*,I6)
      GO TO 10
  100 CONTINUE
      ENDFILE 2
      ENDFILE 3
   45 REWIND 2
      REWIND 3
      WRITE(4,9000)
 9000 FORMAT(1H1,*LISTING OF FILE LEVEL 1*)
  101 READ(3,99) (NPY,(IC(K),K=1,4))
   90 IF(EOF(3)) 103,102
   99 FORMAT(I2,2X,4I6)
  102 WRITE(4,9001) (NPY,(IC(K),K=1,4))
 9001 FORMAT(1H0,I2,2X,4I6)
      GO TO 101
   55 103 READ(2,98)  R,(A(K),K=1,20))
```



```
   98 FORMAT(I6.4X,20A4)
      IF(EOF(2)) 105,104
  104 WRITE(4,9002) (IR,(A(K),K=1,20))
 9002 FORMAT(1H ,I6,4X,20A4)
   60 GO TO 103
  105 CONTINUE
      END
```

```
      PROGRAM DSTIN(INPUT,OUTPUT,DCAT1,DFIL1,TAPE1=INPUT,TAPE4=OUTPUT,
     1TAPE2=DFIL1,TAPE3=DCAT1)
      DIMENSION IC(4),A(20)
      WRITE(4,800)
  05  800 FORMAT(1H1,ERROR LISTING)
      NPY=NPC=IR=NPW=LOG=0
    1 READ(1,90) (NC,NY,NW,(A(K),K=1,18))
   90 FORMAT(10X,I1,I2,I5,18A4)
      IF(NPY-NY) 3,4,50                                    B 4
  10    3 IF(LOG.EQ.0) GO TO 5
      WRITE(3,900) (NPY,(IC(K),K=1,4))
  900 FORMAT(I2,2X,4I6)                                    DSTIN Version 2
    5 LOG=1
      IF(NY.GT.75) GO TO 100
  15  NPY=NY
      NPC=NPW=0
      DO 6 I=1,4
      IC(I)=0
    6 CONTINUE
  20  4 IF(NPC-NC) 7,9,60
    7 NPC=NC
      NPW=0
    9 IF(NPW-NW) 8,10,70
    8 NPW=NW
  25  IC(NC)=IC(NC)+1
   10 IR=IR+1
      WRITE(2,901) (IR,NC,NY,NW,(A(K),K=1,18))
  901 FORMAT(I6,4X,I1,I2,I5,18A4)
      GO TO 1
  30  50 LR=IR+1
      WRITE(4,902) LR
  902 FORMAT(1HO,*YEAR NUMBER CARD*,I6)
      GO TO 10
   60 LR=IR+1
  35  WRITE(4,903) LR
  903 FORMAT(1HO,*CARD TYPE CARD*,I6)
      GO TO 10
   70 LR=IR+1
      WRITE(4,904) LR
  40  904 FORMAT(1HO,*WELL NUMBER CARD*,I6)
      GO TO 10
  100 CONTINUE
      ENDFILE 2
      ENDFILE 3
  45  REWIND 2
      REWIND 3
      WRITE(4,9000)
 9000 FORMAT(1H1,*LISTING OF FILE LEVEL 1*)
  101 READ(3,99) (NPY,(IC(K),K=1,4))
  50  IF(EOF(3)) 103,102
   99 FORMAT(I2,2X,4I6)
  102 WRITE(4,9001) (NPY,(IC(K),K=1,4))
 9001 FORMAT(1HO,I2,2X,4I6)
      GO TO 101
  55  103 READ(2,98) (IR,(A(K),K=1,20))
```



```
   98 FORMAT(I6,4X,20A4)
      IF(EOF(2)) 105,104
  104 WRITE(4,9002) (IR,(A(K),K=1,20))
 9002 FORMAT(1H ,I6,4X,20A4)
  60  GO TO 103
  105 CONTINUE
      END
```

ERROR LISTING

YEAR NUMBER CARD 2

YEAR NUMBER CARD 3

YEAR NUMBER CARD 4

YEAR NUMBER CARD 5

YEAR NUMBER CARD 6

YEAR NUMBER CARD 7

YEAR NUMBER CARD 8

YEAR NUMBER CARD 9

YEAR NUMBER CARD 10

YEAR NUMBER CARD 11

YEAR NUMBER CARD 12

CARD TYPE CARD 13

CARD TYPE CARD 14

CARD TYPE CARD 15

CARD TYPE CARD 16

CARD TYPE CARD 17

CARD TYPE CARD 18

CARD TYPE CARD 19

WELL NUMBER CARD 20

WELL NUMBER CARD 21

WELL NUMBER CARD 22

WELL NUMBER CARD 23

B 5

DSTIN

LISTING OF FILE LEVEL 1

63 0 0 0 7

66 0 0 0 4

67 0 7 15 0

B 6

DSTIN

```
    1   463 1043 2 000  42 152030 256 295030 410 479030 524 606030 611 699030 680 769 11
    2   463 1043 2 030 734 824030 780 870030 815 907030 848 939030 871 965             20
    3   463 1043 3 000 963104803010951162030111611860301129120403011411215030114712 25 11
    4   463 1043 3 030115412330301160123903011641244030116812470301168124 9           20
    5   463 1331 2 000297129840603076309106030763093060307630930603076309306030763093 11
    6   463 1331 2 060307930960603082309606030823096060308530960603085309 6           20
    7   463 1331 7 000 421 419120307330821203086309012030913093120309130931203091309 3 11
    8   466 4216 1 170187718691801894188518019071897180191519061801923191418019291920 11
    9   466 4216 1 180193419251801937192918019411932180194419 34                       20
   10   466 4216 2 030153515420601535154206015361542060153615420601536154206015371542 11
   11   466 4216 2 060153715420601538154206015381542060153815 42                       20
   12   267 4268 1  2870 3000 2843 2996                    315        383 15 60 60 60  10
   13   267 4268 4  4784 4795 4771 4791                    325        383 17 60 21 60  10
   14   267 4277 3  1849 1890 1830 1886                    316        383 15 58 31 54  10
   15   267 4277 4  2727 2660 2710 2858                    319        383 19 31        10
   16   267 4277 5  4355 4411 4342 4408                    322        383 11 25 28  7  10
   17   267 4277 6  3889 3936 3869 3932                    330        383 11 60 60 60  10
   18   267 4277 7  1390 1633 1376 1630                    317        383 10 60 60 60  10
   19   367 4268 1 000 807 862060109211400601115116406011261175060113111830601136118 5 11
   20   367 4268 1 060113911910601142119406011441194060114411940601144119 4           20
   21   367 4268 4 000 361 362060164117280601754180606018081852060184118820601866190 1 11
   22   367 4268 4 060188419140601895192206019061930060191719380601920194 1           20
   23   367 4277 3 000 164 186040 654 676040 694 716040 712 735040 720 746040 726 751 11
   24   367 4277 3 040 731 754040 734 757040 734 759040 734 759040 736 762             20
   25   367 4277 4 00025
   26   367 4277 4 000 251 322030 650 662030 931100 8030 974104603010031073030102110 91 11
   27   367 4277 4 030103411050301045111603010521124030106011320301063113 4           20
   28   367 4277 5 000  11  53025 532 609025 932101002512181260027139214330251516155 6 11
   29   367 4277 5 025159416400251653169302516911732025171817610251739178 0           20
   30   367 4277 6 000  29  51060 338 349060 509 530060 636 646060 726 743060 799 816 11
   31   367 4277 6 060 860 873060 910 927060 952 970060 989100806010211040             20
   32   367 4277 7 000 116 237060 508 609060 541 641060 549 649060 554 652060 554 654 11
   33   367 4277 7 060 557 654060 557 657060 559 657060 559 657060 559 657             20
```

```
** CYBER76 SCOPE 2.0 LVL 65  30/10/73  17.57.08   07/11/73      73311

CSIRO DA03.30/SCOPE2.0 CSIRONET SYSTEM MOD65D.

HH.MM.SS CPU SECOND ORIGIN
          10.51.21 STN.      DIOS  VERSION 24/10/73.
          10.51.21 STN.          91 RECORDS READ.                                    B 7
10.51.23 00000.004 JOB.     -DCARME(T30,P7000)
10.51.23 00000.005 JOB.     -ACCOUNT(CCDMR*WC)
10.51.23 00000.005 JOB.     -ATTACH(DCAT1,DCAT1,ID=CCDMR*WC)
10.51.23 00000.008 SYS.        PF673 - CYCLE     1 ATTACHED                         DCARME
10.51.23 00000.008 JOB.     -ATTACH(DFIL1,DFIL1,ID=CCDMR*WC)
10.51.23 00000.011 SYS.        PF673 - CYCLE     1 ATTACHED
10.51.23 00000.011 JOB.     -ATTACH(DCAT2,DCAT2,ID=CCDMR*WC)
10.51.24 00000.014 SYS.        PF673 - CYCLE     2 ATTACHED
10.51.24 00000.014 JOB.     -ATTACH(DFIL2,DFIL2,ID=CCDMR*WC)
10.51.24 00000.017 SYS.        PF673 - CYCLE     2 ATTACHED
10.51.24 00000.017 LOD.     -REWIND(DCAT1,DFIL1,DCAT2,DFIL2)
10.51.24 00000.020 LOD.     -FTN.
10.51.26 00000.303 USR.           .274 CP SECONDS COMPILATION TIME
10.51.26 00000.303 LOD.     -LGO.
10.51.26 00000.335 SYS.        LD610 - FLS REQUIRED TO LOAD = 0006450 OU.COR
10.51.26 00000.336 SYS.        LD603 - EXECUTION INITIATED 05.EXP
10.51.26 00000.336 USR.       FORTRAN LIBRARY 65            24/08/73
10.51.27 00000.999 USR.           END  DCARME
10.51.28 00001.001 USR.           .663 CP SECONDS EXECUTION TIME
10.51.28 00001.002 JOB.     -PURGE(DCAT2)
10.51.28 00001.003 SYS.        PF256 - FILE SUCCESSFULLY PURGED
10.51.28 00001.003 JOB.     -PURGE(DFIL2)
10.51.28 00001.005 SYS.        PF256 - FILE SUCCESSFULLY PURGED
10.51.28 00001.005 LOD.     -UNLOAD(DCAT2,DFIL2)
10.51.29 00001.009 JOB.     -CATALOG(DCAT3,DCAT2,CY=2,ID=CCDMR*WC,RP=30)
10.51.29 00001.011 SYS.        PF060 - CYCLE     2 CATALOGED
10.51.29 00001.012 JOB.     -CATALOG(DFIL3,DFIL2,CY=2,ID=CCDMR*WC,RP=30)
10.51.29 00001.014 SYS.        PF060 - CYCLE     2 CATALOGED
10.51.29 00001.014 JOB.     -PURGE(DCAT1)
10.51.30 00001.015 SYS.        PF256 - FILE SUCCESSFULLY PURGED
10.51.30 00001.016 JOB.     -PURGE(DFIL1)
10.51.30 00001.017 SYS.        PF256 - FILE SUCCESSFULLY PURGED
10.51.30 00001.018 JOB.     -EXIT.
10.51.30 00001.021 SYS.        RM770 - MAXIMUM ACTIVE FILES              5
10.51.30 00001.021 SYS.        RM771 - OPEN/CLOSE CALLS                 45
10.51.30 00001.021 SYS.        RM772 - DATA TRANSFER CALLS           3,143
10.51.30 00001.021 SYS.        RM773 - CONTROL/POSITIONING CALLS        59
10.51.30 00001.021 SYS.        RM774 - BM DATA TRANSFER CALLS          348
10.51.30 00001.022 SYS.        RM775 - BM CONTROL/POSITIONING CALLS     82
10.51.30 00001.022 SYS.        RM776 - QUEUE MANAGER CALLS              71
10.51.30 00001.022 SYS.        RM777 - RECALL CALLS                    103
10.51.30 00001.023 SYS.        SCM                  8.659  KWS
10.51.30 00001.023 SYS.        I/O                  0.022  MS
10.51.30 00001.023 SYS.        RMS                  0.011  MWS
10.51.30 00001.023 SYS.        USER                 0.454  SEC
10.51.30 00001.023 SYS.        JOB                  1.025  SEC
10.51.30 00001.024 SYS.        SC050 - 000000 SC/LC SWAPS
10.51.30 00001.024 SYS.        PRI                     7
10.51.30 00001.024 SYS.        URC                  0.210  SEC
10.51.30 00001.024 SYS.        TOT                  1.817  SEC
```

```
      PROGRAM DCARME(INPUT,OUTPUT,DCAT1,DCAT2,DCAT3,DFIL1,DFIL2,DFIL3,
     1TAPE61=OUTPUT,TAPE4=DCAT1,TAPE5=DFIL1,TAPE6=DCAT2,TAPE7=DFIL2,
     2TAPE8=DCAT3,TAPE9=DFIL3)
      DIMENSION IYR(2),IC(2,4),ID(4)
      WRITE(61,92)
   92 FORMAT(1H1,25HCATALOGUE OF LEVEL 2 FILE)
   91 FORMAT(X,I2,2X,4I6)
      IR=0                                                    B 8
    1 READ(4,90) (IYR(1),(IC(1,I),I=1,4))
   90 FORMAT(I2,2X,4I6)
      IF(EOF(4)) 3,2                                          DCARME
    3 IYR(1)=99
    2 READ(6,90) (IYR(2),(IC(2,I),I=1,4))
      IF(EOF(6)) 5,4
    5 IYR(2)=99
    4 IF(IYR(1)-IYR(2)) 6,7,8
    8 WRITE(8,90) (IYR(2),(IC(2,I),I=1,4))
      WRITE(61,91) (IYR(2),(IC(2,I),I=1,4))
      M=IC(2,1)+IC(2,2)+IC(2,3)+IC(2,4)
      CALL DWRI(7,5,M,0,IR)
      GO TO 2
    7 IF(IYR(1).EQ.99) GO TO 100
      DO 9 K=1,4
      IF(IC(1,K).LE.0) GO TO 10
      ID(K)=IC(1,K)
      CALL DWRI(5,7,ID(K),IC(2,K),IR)
      GO TO 9
   10 IF(IC(2,K).LE.0) GO TO 11
      ID(K)=IC(2,K)
      CALL DWRI(7,5,ID(K),IC(1,K),IR)
      GO TO 9
   11 ID(K)=0
    9 CONTINUE
      WRITE(8,90) (IYR(1),(ID(K),K=1,4))
      WRITE(61,91) (IYR(1),(ID(K),K=1,4))
      GO TO 1
    6 M=IC(1,1)+IC(1,2)+IC(1,3)+IC(1,4)
      CALL DWRI(5,7,M,0,IR)
      WRITE(8,90) (IYR(1),(IC(1,I),I=1,4))
      WRITE(61,91) (IYR(1),(IC(1,I),I=1,4))
      READ(4,90) (IYR(1),(IC(1,I),I=1,4))
      IF(EOF(4)) 12,4
   12 IYR(1)=99
      GO TO 4
  100 CONTINUE
      END
```

```
** CYBER76 SCOPE 2.1 LVL 65  39/10/73  17.57.08   07/11/73      73311

CSIRC CA83,60/5?OPE2.0 CSIPUNET SYSTEM M0865D.

HH,MP,SS CPU SECOND ORIGIN
            11,51.55 SYS,       IOS  VERSION 24/10/73.                                        R 9
            11,51.56 SYS,         10? RECORDS READ.
10,51,57 00000,004 JOB,     -DUMPGL(T100,P1000)
10,51,57 00000,005 JOB,     -ACCOUNT(CCDMR*WC)                                                DMERGE
10,51,57 00000,005 JOB,     -STAGE(DLEV3,MT,POST,HY,ST=IOS)
10,51,57 00000,006 JOB,     -FILE(DLEV3,MT=W,BT=I,MRL=5120,CM=NO)
10,51,57 00000,007 JOB,     -LABEL(DLEV3,L=$DSTAP$,T=10)
10,51,57 00000,008 JOB,     -STAGE(DLEV2,MT,PRE,HY,ST=IOS,VSN=   9)
10,51,57 00000,009 JOB,     -LABEL(DLEV2,R,L=$DSTAP$)
10,51,57 00000,009 JOB,     -FILE(DLEV2,RT=W,BT=I,MRL=5120,CM=NO)
10,51,57 00000,010 LOD,     -REWIND(DLEV2)
10,51,57 00000,013 JOB,     -ATTACH(DCAT2,DCAT2,ID=CCDMR*WC)
10,51,57 00000,015 SYS,        PF673 - CYCLE     2 ATTACHED
10,51,57 00000,016 JOB,     -ATTACH(DFIL2,DFIL2,ID=CCDMR*WC)
10,51,58 00000,018 SYS,        PF673 - CYCLE     2 ATTACHED
10,51,58 00000,019 LOD,     -COPYP(DLEV2,DCAT3)
10,51,58 00000,022 SYS,        JM260 - STAGE MT IN  LFN=DLEV2   VSN=000009 CC=CCDMR*WC
            11,24.49 IOS,       --  MT00   WRONG VSN
            11,24.51 IOS,       --  MT00   GO
            11,24.51 IOS,       --  MT00   EXPIRED
            11,24.59 IOS,        JM511 - WORDS  READ-         2567
11,30,00 00000,028 USR,          UT030 - COPY COMPLETE
11,30,00 00000,029 USR,          UT034 -      EOR = 7              EOS =          0    EOF = 1
11,30,00 00000,029 LOD,     -COPYP(DLEV2,DFIL3)
11,30,00 00000,079 USR,          UT030 - COPY COMPLETE
11,30,00 00000,080 USR,          UT034 -      EOR = 148            EOS =          0    EOF = 1
11,30,00 00000,080 LOD,     -REWIND(DCAT2,DCAT3,DFIL2,DFIL3)
11,30,00 00000,084 LOD,     -FTN.
11,30,06 00000,377 USR,           ,284 CP SECONDS COMPILATION TIME
11,30,06 00000,378 LOD,     -LGO.
11,30,06 00000,411 SYS,        LD610 - FLS REQUIRED TO LOAD = 0006464 OU,COG
11,30,07 00000,412 SYS,        LD603 - EXECUTION INITIATED OS,EXP
11,30,07 00000,412 USR,       FORTRAN LIBRARY 65          24/08/73
11,30,09 00001,056 USR,           END  DMERGE
11,30,09 00001,059 USR,           ,643 CP SECONDS EXECUTION TIME
11,30,09 00001,059 LOD,     -REWIND(DCAT4,DFIL4?
11,30,09 00001,062 LOD,     -COPY(DCAT4,DLEV3)
11,30,09 00001,071 USR,          UT031 - EOI ENCOUNTERED
11,30,09 00001,071 USR,          UT034 -      EOR = 10             EOS =          0    EOF = 1
11,30,09 00001,072 LOD,     -COPY(DFIL4,DLEV3)
11,30,11 00001,326 USR,          UT031 - EOI ENCOUNTERED
11,30,11 00001,326 USR,          UT034 -      EOR = 850            EOS =          0    EOF = 1
11,30,11 00001,327 LOD,     -REWIND(DCAT2,DFIL2)
11,30,11 00001,330 LOD,     -COPY(DCAT2,DCAT2R)
11,30,11 00001,336 USR,          UT031 - EOI ENCOUNTERED
11,30,11 00001,337 USR,          UT034 -      EOR = 10
11,30,11 00001,337 LOD,     -COPY(DFIL2,DFIL2R)
11,30,12 00001,452 USR,          UT031 - EOI ENCOUNTERED
11,30,12 00001,453 USR,          UT034 -      EOR = 850
11,30,12 00001,454 JOB,     -CATALOG(DCAT2R,DCAT2R,ID=CCDMR*WC,RP=15)
11,30,12 00001,455 SYS,        PF030 - PF NAME ALREADY IN SYSTEM
11,30,12 00001,508 SYS,        SC031 - JOB ABORTED
11,30,13 00001,511 JOB,     -EXIT.
11,30,13 00001,512 JOB,     -ATTACH(ERCAT,DCAT2R,ID=CCDMR*WC)
11,30,13 00001,514 SYS,        PF673 - CYCLE     1 ATTACHED
11,30,13 00001,515 JOB,     -ATTACH(ERFIL,DFIL2R,ID=CCDMR*WC)
11,30,14 00001,517 SYS,        PF673 - CYCLE     1 ATTACHED
11,30,13 00001,519 JOB,     -PURGE(ERCAT)

11,30,14 00001,519 SYS,        PF256 - FILE SUCCESSFULLY PURGED
11,30,14 00001,519 JOB,     -PURGE(ERFIL)
11,30,14 00001,521 SYS,        PF256 - FILE SUCCESSFULLY PURGED
11,30,14 00001,521 LOD,     -UNLOAD(ERCAT,ERFIL)
11,30,14 00001,524 JOB,     -CATALOG(DCAT2R,DCAT2R,ID=CCDMR*WC,RP=15)
11,30,15 00001,527 SYS,        PF060 - CYCLE     1 CATALOGED
11,30,15 00001,528 JOB,     -CATALOG(DFIL2R,DFIL2R,ID=CCDMR*WC,RP=15)
11,30,16 00001,530 SYS,        PF060 - CYCLE     1 CATALOGED
11,30,16 00001,530 JOB,     -PURGE(DCAT2)
11,30,16 00001,532 SYS,        PF256 - FILE SUCCESSFULLY PURGED
11,30,16 00001,532 JOB,     -PURGE(DFIL2)
11,30,16 00001,533 SYS,        PF256 - FILE SUCCESSFULLY PURGED
11,30,16 00001,535 SYS,        JM261 - STAGE MT OUT LFN=DLEV3   VSN=SCRTCH CC=CCDMR*WC
            11,34.10 IOS,      TAE NUMBER 3194 ASSIGNED
            11,34.16 IOS,       --  MT00   LABEL PARITY ERROR
            11,34.16 IOS,       --  MT00   LABEL PARITY ERROR
            11,34.16 IOS,       --  MT00   UNLABELLED
            11,34.21 IOS,       --  MT00   GO
            11,34.26 IOS,        JM513 - WORDS  WRITTEN-     17113
11,34,28 00001,541 SYS,          RM770 - MAXIMUM ACTIVE FILES                    11
11,34,28 00001,541 SYS,          RM771 - OPEN/CLOSE CALLS                        53
11,34,28 00001,542 SYS,          RM772 - DATA TRANSFER CALLS                  7,117
11,34,28 00001,542 SYS,          RM773 - CONTROL/POSITIONING CALLS               83
11,34,28 00001,542 SYS,          RM774 - BM DATA TRANSFER CALLS                 539
11,34,28 00001,542 SYS,          RM775 - BM CONTROL/POSITIONING CALLS           119
11,34,28 00001,543 SYS,          RM776 - QUEUE MANAGER CALLS                    113
11,34,28 00001,543 SYS,          RM777 - RECALL CALLS                           136
11,34,28 00001,543 SYS,          SCM                  9.712  KWS
11,34,28 00001,543 SYS,          I/O                  0.059  MW
11,34,28 00001,544 SYS,          MMS                  0.030  MWS
11,34,28 00001,544 SYS,          USER                 0.657  SEC
11,34,28 00001,544 SYS,          JOB                  1.546  SEC
11,34,28 00001,544 SYS,          SC050 - 000010 SC/LC SWAPS
11,34,28 00001,545 SYS,          PRI                     1
11,34,28 00001,545 SYS,          URC                  5.144  SEC
11,34,28 00001,545 SYS,          TUT                  7.831  SEC
```

```
      PROGRAM DMERGE(INPUT,OUTPUT,DCAT2,DCAT3,DCAT4,DFIL2,DFIL3,DFIL4,
     1TAPE61=OUTPUT,TAPE4=DCAT2,TAPE6=DCAT3,TAPE8=DCAT4,TAPE5=DFIL2,
      TAPE7=DFIL3,TAPE9=DFIL4)
      DIMENSION IYR(2),IC(2,4),ID(4)
      WRITE(61,92)
   91 FORMAT(X,I2,2X,4I6)
   92 FORMAT(1H1,,5HCATALOGUE OF LEVEL 3 FILE)
      IR=0
    1 READ(4,90) (IYR(1),(IC(1,I),I=1,4))
   90 FORMAT(I2,2X,4I6)
      IF(EOF(4)) 3,2
    3 IYR(1)=99
    2 READ(8,90) (IYR(2),(IC(2,I),I=1,4))
      IF(EOF(8)) 5,4
    5 IYR(2)=99
    4 IF(IYR(1)-IYR(2)) 6,7,8
    8 WRITE(8,90) (IYR(2),(IC(2,I),I=1,4))
      WRITE(61,91) (IYR(2),(IC(2,I),I=1,4))
      M=IC(2,1)+IC(2,2)+IC(2,3)+IC(2,4)
      CALL DWRI(7,5,M,0,IR)
      GO TO 2
    7 IF(IYR(1).EQ.99) GO TO 100
      DO 9 K=1,4
      IF(IC(1,K).LE.0) GO TO 10
      ID(K)=IC(1,K)
      CALL DWRI(5,7,ID(K),IC(2,K),IR)
      GO TO 9
   10 IF(IC(2,K).LE.0) GO TO 11
      ID(K)=IC(2,K)
      CALL DWRI(7,5,ID(K),IC(1,K),IR)
      GO TO 9
   11 ID(K)=0
    9 CONTINUE
      WRITE(8,90) (IYR(1),(ID(K),K=1,4))
      WRITE(61,91) (IYR(1),(ID(K),K=1,4))
      GO TO 1
    6 M=IC(1,1)+IC(1,2)+IC(1,3)+IC(1,4)
      CALL DWRI(5,7,M,0,IR)
      WRITE(8,90) (IYR(1),(IC(1,I),I=1,4))
      WRITE(61,91) (IYR(1),(IC(1,I),I=1,4))
      READ(4,90) (IYR(1),(IC(1,I),I=1,4))
      IF(EOF(4)) 12,4
   12 IYR(1)=99
      GO TO 4
  100 CONTINUE
      ENDFILE 9
      ENDFILE 8
      END
```



```
      SUBROUTINE DWRI(IN,LN,N,M,IR)
      DIMENSION A(20)
      DO 1 L=1,N
      IF(IN.EQ.5) GO TO 6
      READ(IN,90) (A(K),K=1,20)
      GO TO 7
    6 READ(IN,91) (ID,(A(K),K=1,20))
    7 WRITE(9,90) (A(K),K=1,20)
    1 CONTINUE
   90 FORMAT(20A4)
   91 FORMAT(I6,4X,20A4)
      IF(M.LT.1) GO TO 2
      DO 3 L=1,M
      READ(LN,92) A(1)
   92 FORMAT(A4)
    3 CONTINUE
    2 CONTINUE
      RETURN
      END
```

CATALOGUE 0: LEVEL 3 FILE

62	12	0	0	0
63	39	22	44	36
64	56	52	104	74
65	13	21	42	32
66	16	16	28	16
67	7	10	20	15
68	11	13	27	28
69	11	7	14	14
70	11	8	16	14
72	1	0	0	0

```
** CYBER76 SCOPE 2,0 LVL 65  08/11/73  13,20.36   14/11/73      73318

CSIRO DAD3.30/SCOPE2,0 CSIRONET SYSTEM MOD65D,

HH.MM.SS CPU SECOND ORIGIN
          14.57.09 STN.      DIOS  VERSION 24/10/73.
          14.57.09 STN.          136 RECORDS READ.
14.57.11 00000.005 JOB.     -DRETR(T100,P1000)
14.57.11 00000.005 JOB.     -ACCOUNT(CCDMR*WC)
14.57.11 00000.006 JOB.     -STAGE(DLEV2,MT,PRE,HY,ST=IOS,VSN=3194)                    B 12
14.57.11 00000.007 JOB.     -FILE(DLEV2,RT=S,ET=I,MBL=5120,CM=NO)
14.57.11 00000.008 JOB.     -LABEL(DLEV2,R,L=3DSTAPS)
14.57.11 00000.008 JOB.     -FILE(DCAT3)
14.57.11 00000.009 JOB.     -FILE(DFIL3)                                              DRETRI
14.57.11 00000.010 LOD.     -COPYP(DLEV2,DCAT3)
14.57.11 00000.013 SYS.        JM260 - STAGE MT IN  LFN=DLEV2   VSN=003194 CC=CCDMR*WC
          15.33.21 IOS.      -- MT00   WRONG VSN
          15.33.24 IOS.      -- MT00   GO
          15.33.24 IOS.      -- MT00   EXPIRED
          15.33.28 IOS.        JM511 - WORDS  READ=       17113
15.33.29 00000.020 USR.        UT030 - COPY COMPLETE
15.33.29 00000.020 USR.        UT034 -     EOR - 10          EOS -          0    EOP - 1
15.33.29 00000.021 LOD.     -COPYP(DLEV2,DFIL3)
15.33.29 00000.289 USR.        UT030 - COPY COMPLETE
15.33.29 00000.289 USR.        UT034 -     EOR - 850         EOS -          0    EOP - 1
15.33.29 00000.290 LOD.     -FTN.
15.33.31 00000.679 USR.           ,380 CP SECONDS COMPILATION TIME
15.33.31 00000.679 LOD.     -LGO.
15.33.32 00000.715 SYS.        LD610 - FLS REQUIRED TO LOAD = 0006551 OU.COG
15.33.32 00000.716 SYS.        LD603 - EXECUTION INITIATED OS.EXP
15.33.32 00000.716 USR.       FORTRAN LIBRARY 65          24/08/73
15.33.33 00001.445 USR.           END  DRETR
15.33.33 00001.449 USR.           ,731 CP SECONDS EXECUTION TIME
15.33.33 00001.449 JOB.     -CATALOG(DCAR1,D01,ID=CCDMR*WC,RP=7)
15.33.33 00001.452 SYS.        PF060 - CYCLE    1 CATALOGED
15.33.33 00001.453 JOB.     -CATALOG(DCAR2,D02,ID=CCDMR*WC,RP=7)
15.33.34 00001.455 SYS.        PF060 - CYCLE    1 CATALOGED
15.33.34 00001.455 JOB.     -CATALOG(DCAR3,D03,ID=CCDMR*WC,RP=7)
15.33.34 00001.457 SYS.        PF060 - CYCLE    1 CATALOGED
15.33.34 00001.458 JOB.     -CATALOG(DCAR4,D04,ID=CCDMR*WC,RP=7)
15.33.35 00001.460 SYS.        PF060 - CYCLE    1 CATALOGED
15.33.35 00001.465 SYS.        RM770 - MAXIMUM ACTIVE FILES                  11
15.33.35 00001.466 SYS.        RM771 - OPEN/CLOSE CALLS                      54
15.33.35 00001.466 SYS.        RM772 - DATA TRANSFER CALLS                5,759
15.33.35 00001.466 SYS.        RM773 - CONTROL/POSITIONING CALLS             75
15.33.35 00001.466 SYS.        RM774 - BM DATA TRANSFER CALLS               520
15.33.35 00001.466 SYS.        RM775 - BM CONTROL/POSITIONING CALLS          93
15.33.35 00001.467 SYS.        RM776 - QUEUE MANAGER CALLS                   96
15.33.35 00001.467 SYS.        RM777 - RECALL CALLS                         107
15.33.35 00001.467 SYS.        SCM                  11.842  KWS
15.33.35 00001.466 SYS.        I/O                   0.040  MW
15.33.35 00001.468 SYS.        RMS                   0.039  MWS
15.33.35 00001.468 SYS.        USER                  0.748  SEC
15.33.35 00001.468 SYS.        JOB                   1.470  SEC
15.33.35 00001.468 SYS.        SC050 - 000002 SC/LC SWAPS
15.33.35 00001.469 SYS.        PRI                      1
15.33.35 00001.469 SYS.        URC                   2.718  SEC
15.33.35 00001.469 SYS.        TOT                   5.177  SEC
```

```
      PROGRAM DRETR(INPUT,OUTPUT,DCAT3,DFILE3,DCATE,DCAR1,DCAR2,DCAR3,
     1DCAR4,TAPE1=DCAR1,TAPE2=DCAR2,TAPE3=DCAR3,TAPE4=DCAR4,
     2TAPE11=DCAT3,TAPE12=DFILE3,TAPE13=DCATE,TAPE60=INPUT,
     3TAPE61=OUTPUT)
      DIMENSION LU(4),IC(4),A(20)
      CALL INSPEC
      READ(60,90) (LU(K),K=1,4)
   90 FORMAT(4I2)
      REWIND 11
      REWIND 12                                                    B 13
      READ(13,91) IYE,ICE
   91 FORMAT(2I2)
      IF(EOF(13))100,2                                             DRETRI
    2 READ(11,92) IYT,(IC(K),K=1,4)
   92 FORMAT(I2,2X,4I6)
      IF(EOF(11)) 80,3
    3 DO 5 N=1,4
      IF(IC(N),EQ.0) GO TO 5
    4 IF(IYE-IYT) 10,6,30
    6 IF(ICE-IC(N)) 10,20,30
   10 WRITE(61,900) IYE,ICE
  900 FORMAT(8H VOLUME ,I3,6H PAGE ,I3,13H NOT ON TAPE )
      READ(13,91) IYE,ICE
      IF(EOF(13)) 100,4
   20 IF(LU(N),EQ.0) LU(N)=1
      IL=LU(N)
      M=IC(N)
      DO 21 I=1,M
      READ(12,93) (A(K),K=1,20)
   93 FORMAT(20A4)
      WRITE(IL,93) (A(K),K=1,20)
   21 CONTINUE
      READ(13,91) IYE,ICE
      IF(EOF(13)) 100,5
   30 M=IC(N)
      DO 31 I=1,M
      READ(12,93) A(1)
   31 CONTINUE
    5 CONTINUE
      GO TO 2
   80 READ(13,91) IYE,ICE
      IF(EOF(13)) 100,81
   81 WRITE(61,900) IYE,ICE
      GO TO 80
  100 NN=999
      WRITE(1,94) NN
   94 FORMAT(I3)
      ENDFILE 1
      END
```

14587

```
      SUBROUTINE INSPEC
      DIMENSION IS(26),AS(26)
      READ(60,89) C1,C2,C3
   89 FORMAT(3A1)
    1 READ(60,90) (IS(K),AS(K),K=1,26)                                  B 14
   90 FORMAT(26(I2,A1))
   50 N=1
      IF(N.EQ.1.AND.AS(N).EQ.C3) GO TO 100
    2 IF(N.GT.25) GO TO 1
      IF(AS(N).EQ.C3) GO TO 1                                           INSPEC
      IF(AS(N).EQ.C1.AND.AS(N+1).EQ.C1) GO TO 10
      IF(AS(N).EQ.C2.AND.AS(N+2).EQ.C1) GO TO 20
      IF(AS(N+1).EQ.C2.AND.AS(N+2).EQ.C1) GO TO 30
      IF(AS(N).EQ.C2.AND.AS(N+2).EQ.C2) GO TO 40
      WRITE(61,900)
  900 FORMAT(X,28HINCORRECT INPUT CARD SKIPPED )
      GO TO 1
   10 WRITE(13,91) IS(N),IS(N+1)
   91 FORMAT(I2,X,I1)
      N=N+2
      GO TO 2
   20 L1=IS(N)
      L2=IS(N+1)
      DO 21 L=L1,L2
      WRITE(13,91) L,IS(N+2)
   21 CONTINUE
      N=N+3
      GO TO 2
   30 L1=IS(N+1)
      L2=IS(N+2)
      DO 31 L=L1,L2
      WRITE(13,91) IS(N),L
   31 CONTINUE
      N=N+3
      GO TO 2
   40 L1=IS(N)
      L2=IS(N+1)
      K1=IS(N+2)
      K2=IS(N+3)
      DO 41 L=L1,L2
      DO 42 K=K1,K2
      WRITE(13,91) L,K
   42 CONTINUE
   41 CONTINUE
      N=N+4
      GO TO 2
  100 ENDFILE 13
      REWIND 13
      RETURN
      END
```

LOAD MAP LOADER VER. 1.0 PAGE 1

PROGRAM WILL BE ENTERED AT DRETR (410) SCM LENGTH 6606 LCM LENGTH 0

BLOCK	ADDRESS	LENGTH
DRETR	100	643
INSPEC	743	353
LIST	1316	63
ENDFIL=	1401	44
EOF	1445	16
FORSYS=	1463	1140
GETFIT=	2623	33
INPC=	2656	241
KODER=	3117	1456
KRAKER=	4575	1551
OUTC=	6346	203
REWIND=	6551	35

B 15

DRETRI

VOLUME 61 PAGE 1 NOT ON TAPE
VOLUME 61 PAGE 2 NOT ON TAPE
VOLUME 61 PAGE 3 NOT ON TAPE
VOLUME 61 PAGE 4 NOT ON TAPE
VOLUME 62 PAGE 2 NOT ON TAPE
VOLUME 62 PAGE 3 NOT ON TAPE
VOLUME 62 PAGE 4 NOT ON TAPE
VOLUME 71 PAGE 1 NOT ON TAPE
VOLUME 71 PAGE 2 NOT ON TAPE
VOLUME 71 PAGE 3 NOT ON TAPE
VOLUME 71 PAGE 4 NOT ON TAPE

```
162 1082PPOBLACK MTN   14F54 614017  2230    861 5499 4 1 0 0 0
162 1087AAOSUNNYBANK   14G551214913202656 25  832 7131 6 5 0 0 0
162 1098AAOKILLORAN    14G55 71481930255410 1707 2350 3 3 0 0 0
162 1107UKABURUNGA     14G55 915009432600 00 109110242 3 3 0 0 0
162 1108AAOPENRITH     14F541614338302310 20  709 4073 3 3 0 0 0
162 1113AAOPLEASANT H 14G551214900102625 10 1253 3485 5 2 0 0 0
162 1119ODNMARANDA     14F551314526402312 14  862 6491 7 6 0 0 0
162 1122ENLBROOKWOOD   14F55 914419582228 55  750 4806 1 1 0 0 0DRY ABND
162 1214CCLMARINA      14054121435230143810    80 3829 4 3 0 0 0
162 1303AADWESTGROVE   34G55 714826302534 00 173212663 8 4 0 0 0
162 1309UKACROWDER     14G561315015342752 36  868 5864 6 1 0 0 0
162 1311AAOKILDARE     14G55 714824625431 16 1604 5724 9 5 0 0 0
163 1000AAOBINYA       14G551214831202641 57 1290 4318 2   0 0 0
163 1002DSLPANDIEBURR  14G54 913925312645 35  117 7253 1 1 1 2 2
163 1004AAOBONYCREEK   14G551214856802645 00 1033 4583 4 4 0 0  PROD
163 1006UKAWERIBONE    14G551614921022719 38 1115 7995 1 1 0 0 0
163 1014PECWARRINILLA  14G55 814833142506 49 1021 670117   0 0 0PLUGGED + ABND
163 1015AODCANAWAY     14G54 814357472556 06  783 4930 5 5 0 0 0
163 1019AAORICHMOND    14G551214853002640 30  980 4130 1 1 0 0 0
163 1021AADINGLE       14G551214849102638 50 1000 4197 2 1 0 0 0
163 1029DSLGIDGEALPA   15G541414004562756 46 16113114 9 5 0 0 0
163 1031PPCTINKER CK   14G561315047462745 25 1154 4207 2 2 2 4 2ABND
163 1032PPCWAGGARA     14G561315054562742 01 1136 4025 5 4 1 2 2
163 1033PPCZIG ZAG     14G561315055432753 25 1161 2246 1 1 1 2 2
163 1036APOROLLESTON   14G55 414837202434 00  668 9508 14 9 0 0 0
163 1037AADAPPLE GROV 14G551214851062640 55  937 4146 1 1 0 0 0
163 1038AADLOPIE       14G551114826002642 15 1120 4250 3 3 0 0 0
163 1039AADBARDLMING 14G551214910042643 40 1038 4697 2 2 0 0 0
163 1040AADRUSEDALE    14G551214856392654 51  974 5255 4 4 0 0 0ABND
163 1042UKAWONGER      14G551614980342744 45 1005 6339 1 1 0 0 0ABND
163 1043PPCCROILPLAIN 14G561315114502731 57 1178 5501 4 4 3 6 6DRY ABND
163 1044PPCWHARFILLA14G561315047462730 38 1276 4034 2 2 1 2 2
```

```
** CYBER76 SCOPE 2.0 LVL 65  05/11/73  13.20.36   13/11/73      73317

CSIRO EAD3,30/SCOPE2.0 CSIRONET SYSTEM MOD650.

HH,MM,SS CPU SECOND ORIGIN
            14.06.10 STN.      DIOS  VERSION 24/10/73.
            14.06.10 STN.        119 RECORDS READ.
14,06,11 00000,006 JOB.     -DTAPEX(T100,F1000,DAZ00)
14,06,11 00000,006 JOB.     -ACCOUNT(CCDMR*WC)
14,06,11 00000,007 JOB.     -STAGE(DLEV2,MT,PRE=NY,ST=1AS,VSN=3194)            B 16
14,06,11 00000,008 JOB.     -FILE(DLEV2,RT=W,BT=I,MRL=5120,CM=NO)
14,06,11 00000,008 JOB.     -LABEL(DLEV2,R,L=ESDSTAPS)
14,06,11 00000,009 JOB.     -FILE(DCAT3)
14,06,11 00000,010 JOB.     -FILE(DFIL3)                                       DTAPEX
14,06,11 00000,011 LOD.     -COPYP(DLEV2,DCAT3)
14,06,11 00000,014 SYS.        JM260 - STAGE MT IN  LFN=DLEV2   VSN=003194 CC=CCDMR*WC
            14,13.35 IOS.     --  MT01   WRONG VSN
            14,13.37 IOS.     --  MT01   GO
            14,13.37 IOS.     --  MT01   EXPIRED
            14,13.42 IOS.        JM511 - WORDS  READ-        17113
14,13,42 00000,021 USR.        UT030 - COPY COMPLETE
14,13,42 00000,021 USR.        UT034 -     EOR = 10          EOS -         0    EOF = 1
14,13,42 00000,022 LOD.     -COPYP(DLEV2,DFIL3)
14,13,42 00000,266 USR.        UT030 - COPY COMPLETE
14,13,42 00000,268 USR.        UT034 -     EOR = 650         EOS -         0    EOP = 1
14,13,42 00000,269 LOD.     -FIN.
14,13,43 00000,621 USR.          .324 CP SECONDS COMPILATION TIME
14,13,48 00000,622 LOD.     -LGO.
14,13,50 00000,657 SYS.        LD610 - FLS REQUIRED TO LOAD - 0006512 OU,COG
14,13,50 00000,658 SYS.        LD603 - EXECUTION INITIATED OS EXP
14,13,50 00000,658 USR.       FORTRAN LIBRARY 65              24/08/73
14,13,51 00000,767 USR.           END  DREIR
14,13,50 00000,769 USR.          .109 CP SECONDS EXECUTION TIME
14,13,50 00000,770 JOB.     -CATALOG(DCAR1,D01,ID=CCDMR*WC,RP=7)
14,13,51 00000,772 SYS.        PF060 - CYCLE     1 CATALOGED
14,13,51 00000,773 JOB.     -TRANSF(DSTIN3)
            14,13.50 SYS.      JM507 - TRANSFERRED TO NDMR*WC
14,13,51 00000,777 SYS.        RM770 - MAXIMUM ACTIVE FILES                 8
14,13,51 00000,777 SYS.        RM771 - OPEN/CLOSE CALLS                    38
14,13,51 00000,777 SYS.        RM772 - DATA TRANSFER CALLS              3,698
14,13,51 00000,777 SYS.        RM773 - CONTROL/POSITIONING CALLS           54
14,13,51 00000,777 SYS.        RM774 - BM DATA TRANSFER CALLS             376
14,13,51 00000,778 SYS.        RM775 - BM CONTROL/POSITIONING CALLS        69
14,13,51 00000,778 SYS.        RM776 - QUEUE MANAGER CALLS                 73
14,13,51 00000,778 SYS.        RM777 - RECALL CALLS                        73
14,13,51 00000,779 SYS.        SCM                      8.043  KWS
14,13,51 00000,779 SYS.        I/O                      0.024  MW
14,13,51 00000,779 SYS.        MMS                      0.013  MWS
14,13,51 00000,779 SYS.        USER                     0.240  SEC
14,13,51 00000,779 SYS.        JOB                      0.781  SEC
14,13,51 00000,780 SYS.        SC050 - 000001 SC/LC SWAPS
14,13,51 00000,780 SYS.        PRI                          1
14,13,51 00000,780 SYS.        UAC                      2.276  SEC
14,13,51 00000,780 SYS.        TOT                      3.693  SEC
```

17579