JAERI-M
91-201

DATA-POOL : A DIRECT-ACCESS DATA BASE FOR LARGE-SCALE NUCLEAR CODES

December 1991

Naoki YAMANO*, Kinji KOYAMA, Yoshitaka NAITO
and Kazuyoshi MINAMI**

日 本 原 子 力 研 究 所
Japan Atomic Energy Research Institute

JAERI-Mレポートは，日本原子力研究所が不定期に公刊している研究報告書です。

入手の問合わせは，日本原子力研究所技術情報部情報資料課（〒319－11茨城県那珂郡東海村）あて，お申しこしください。なお，このほかに財団法人原子力弘済会資料センター（〒319-11茨城県那珂郡東海村日本原子力研究所内）で複写による実費頒布をおこなっております。

JAERI-M reports are issued irregularly.

Inquiries about availability of the reports should be addressed to Information Division, Department of Technical Information, Japan Atomic Energy Research Institute, Tokai-mura, Naka-gun, Ibaraki-ken 319-11, Japan.

© Japan Atomic Energy Research Institute, 1991

編集兼発行　日本原子力研究所
印　　　刷　日立高速印刷株式会社

DATA-POOL : A Direct-access Data Base for Large-scale Nuclear Codes

Naoki YAMANO*, Kinji KOYAMA, Yoshitaka NAITO
and Kazuyoshi MINAMI**

Department of Fuel Safety Research
Tokai Research Establishment
Japan Atomic Energy Research Institute
Tokai-mura, Naka-gun, Ibaraki-ken

(Received November 5, 1991)

A direct-access data base DATA-POOL has been developed for large-scale nuclear codes. The data can be stored and retrieved with specifications of simple node names, by using the DATA-POOL access package written in the FORTRAN 77 language. A management utility POOL for the DATA-POOL is also provided. A typical application of the DATA-POOL is shown to the RADHEAT-V4 code system developed for performing safety analyses of radiation shielding. Many samples and error messages are also noted to apply the DATA-POOL for the other code systems.

This report is provided for a manual of DATA-POOL.

Keywords : DATA-pool, POOL, Computer Code, Software Package, Direct-access, Data Base, Data Handling, RADHEAT-V4, VISUAL, Nuclear Code, Manual

* Sumitomo Atomic Energy Ind., Ltd.

** Fujitsu Limited.

DATA－POOL：大規模原子力コード用直接編成データベース

日本原子力研究所東海研究所燃料安全工学部
山野　直樹*・小山　謹二・内藤　俶孝・南　　多善**

（1991年11月5日受理）

直接編成ファイルを用いたデータベースDATA－POOLを大型原子力コードのために開発した。データは簡単なノード名の指定によって格納・検索される。DATA－POOL 処理パッケージはFORTRAN 77 言語で作成されている。保守管理ユーティリティPOOLも併せて用意されている。DATA－POOLの典型的な応用例として，放射線遮蔽安全解析コードシステムRADHEAT－V4への適用を示した。DATA－POOLを他のシステムに適用する為の多くの使用例及びエラーメッセージについても述べている。本報告書はDATA－POOLの使用手引書である。

東海研究所：〒319－11　茨城県那珂郡東海村白方字白根2－4
＊　住友原子力工業（株）
＊＊　富士通（株）

Contents

目　次

1. Introduction

For estimating radiation damage and dose rate in shielding safety evaluation, precise calculations have been performed by using large-size multi-group transport codes according to progress of computer resource. An amount of data on cross sections and radiation distributions becomes so large and the structure of them becomes also so complex that it is not easy to treat them using the conventional sequential data form. To avoid the difficulty and execute the effective processing, a direct-access data base DATA-POOL has been developed for the large-size nuclear codes.

The basic concept of DATA-POOL was designed by the JAERI Nuclear Code Committee. A DATAPOOL code[1)] was developed at the JAERI Computing Center in 1980. This DATAPOOL is designed for an general data base that is possible to store many node names and contains an intelligent algorithm for information retrieval by using a LRU (Least Recent Use) table and a DT (Directory Table) buffer. Arbitrary relations of node names can be defined by the user. The DATAPOOL code, however, has a pre-compilation process to translate the CJ statement, which is similar to the FORTRAN language, to the ordinary FORTRAN statements and the program debugging is often trouble some.

As for shielding calculation, a large amount of data is required, however the number of classified node name is not so great. Therefore, the direct-access data base with the large directory retrieval is not necessary to treat the cross sections or radiation distributions. Present DATA-POOL access package which is different from the DATAPOOL code is therefore developed to treat effectively the cross sections and the other related data.

The DATA-POOL access package is characterized as a simplified version of the DATAPOOL code and reduces I/O access times by half compared with the DATAPOOL code to retrieve the same data due to a simple and an appropriate forms of directories. The minimum procedure is required to find the specified node name and the data. The DATA-POOL access package is suitable to treat data of large size with a node name.

The access package consists of several subroutines written by the FORTRAN 77 language. An ASSEMBLER subroutine is only used to get DCB (Data Control Block) information in the DD (Data Definition) statement. The user can use the direct-access data base to assign the load module of the DATA-POOL access package as a private library for the linkage editor.

The DATA-POOL has been adopted to the RADHEAT-V4[3)] code system developed for shielding safety evaluation. The experience of using the access package about 5 years shows that the DATA-POOL is operated regularly. The default value of a physical value of a physical record length is set to 3600 bytes (900 words), however it can be easy to change the appropriate value according to user's system.

The data in the DATA-POOL are classified and identified by a node name consisted of 4-characters. On the other hand, the data are stored and retrieved by the node name specification. The structure of the DATA-POOL is described in Chap. 2. The access package of the DATA-POOL and the utilization are described in Chapters 3 and 4, respectively.

A management utility POOL operated with TSS terminals is prepared for the management of DATA-POOL. The utility has 14 functions such as copy, delete, condense, backup and rename. The usage of the utility POOL is described in Chap. 5. To show the application of DATA-POOL, the special data forms used in the RADHEAT-V4 code system are described in Chap. 6. The data in the DATA-POOL can be plotted as two-, three-dimensional graphs and contour-line maps by using a plotting utility VISUAL[4].

The error messages printed by the access subroutines are noted in Appendix A. The source lists of the access subroutines and of the utility POOL are shown in Appendices B and C, respectively.

2. Structure of DATA-POOL

DATA-POOL is a direct access data set defined by the direct access read/write statements of the FORTRAN 77 language. The data set has a fixed and an unblocked record length of 3600 bytes (900 words).

DATA-POOL consists of three sections named a "Control Section", a "Directory Section" and a "Data Section" as shown in Fig. 2.1. Arbitrary data are stored/retrieved in the DATA-POOL by a standardized format, so that the management of the data can be easily achieved. In this chapter, the concept and the structure of the DATA-POOL are described. The management and plotting ulilities related to the DATA-POOL are briefly described in Section 2.1.

2.1 Concept of DATA-POOL

In the DATA-POOL, the data are labelled by an arbitrary node name which consists of 4-characters defined by the user, and stored in the Data Section. The node name is related to the others and a tree structure is generated as shown in Fig. 2.2.

In this figure, EGRP means an energy group structure commonly used in the data. SGRX means the attribute of secondary gamma-ray production cross sections. INFX means the attribute of infinite dilution cross sections. ELA means the attribute of elastic scattering matrices. The node name frequently referred in the system should be located at the upper level of the tree structure. Data belonging to the same category should be combined and attributed to a node name in order to improve the efficiency of data retrieval.

The tree structure and the node name are stored in the Directory Section together with the direct-access record addresses. The data retrieval is carried out from the node of the first level to that of the lower level, so that the most suitable tree structure is essentially needed according to the property of the data.

The data retrieval for the DATA-POOL is carried out by setting a series of the node names in PFIND/PSET subroutines, and then the data access is carried out by PREAD/PRITE subroutines.

The DATA-POOL adopts an exclusive control for the write access in order to prevent the destruction of DATA-POOL from plural job access. In the period executing between PWSTAT subroutine and PWEND subroutine, the write access by the other jobs is inhibited. These functional subroutines are described in Chap. 3.

The management of the DATA-POOL is carried out by using an utility program POOL. POOL has 14 functions to maintain the data in the DATA-POOL such as initialize, rename, copy, backup. These functions can be executed with TSS terminals or batch jobs. The data in the DATA-POOL can be displayed on TSS graphic terminals (TEKTRONIX T4014) or NLP (Nihongo Line Printer) by using a plotting utility program of VISUAL. Various plottings such as two-dimentional, contour-line and three-dimentional graphs are produced by using the conversational input data. The relation between the utilities and the DATA-POOL is shown in Fig. 2.3.

2.2 Control Section

The Control Section is located at the first record of DATA-POOL and has a size of 40 words. The variables in the section are used for the control of the DATA-POOL. The record structure of the Control Section is shown in Fig. 2.4. The Control Section should be contained in a physical record length of the DATA-POOL. The initialization of the Control Section is carried out by calling PINIT subroutine.

2.3 Directory Section

The Directory Section takes an important role which determines the relation between the node name and the record address of the data, and has the information of the lower nodes in the tree structure. The Directory Section consists of the several sub-directories. The sub-directory has the information of a node name, a head address of the data section associated with the node name, the date of creation and control variables defined by the user. The structure of the Directory Section is shown in Fig. 2.5. A sub-directory takes 12 words in the Directory Section and a directory holds a physical record length, so that the maximum number of nodes associated with the same level is limited to the next value.

$$N_{max} = \mathrm{tranc}\left\{ \frac{(\text{1 physical record length (words)} - 4)}{12} \right\}$$

The DATA-POOL of RADHEAT-V4 has a physical record length of 3600 bytes (900 words), so that the value of N_{max} is 74.

A feature of the DATA-POOL is the information for the nodes of the lower level can be obtained at once by referring a directory, however the DATA-POOL will not be adequate for systems with tree structures contained too many lower nodes.

In the sub-directory, 5 kinds of information can be recorded by the user. The record area is prepared for the reason which the DATA-POOL consists of many kind of data associated with a node name, and will be used as control flags whether data are in existence or nonexistence in the Data Section. The needless access to the data files can be prevented by utilizing the information in the sub-directory.

2.4 Data Section

The Data Section consists of several sub-data sets. A sub-data set is created by executing a writing. The writing is carried out by calling a subroutine of PRITE - PRITE4. The subroutine PRITE only creates the comments of the node. The data of one-dimensional array are written in the regions from DATA1 to DATA4 by calling a subroutine of PRITE1 - PRITE4, respectively. (see Fig. 2.6) The subroutine PRITE1 creates the comments of the node and the DATA1. The subroutine PRITE4 generates the comments of the node, DATA1, DATA2, DATA3 and DATA4 as shown in Fig. 2.6. These data can be read by using the subroutines of PREAD - PREAD4 which correspond to the PRITE - PRITE4 subroutines.

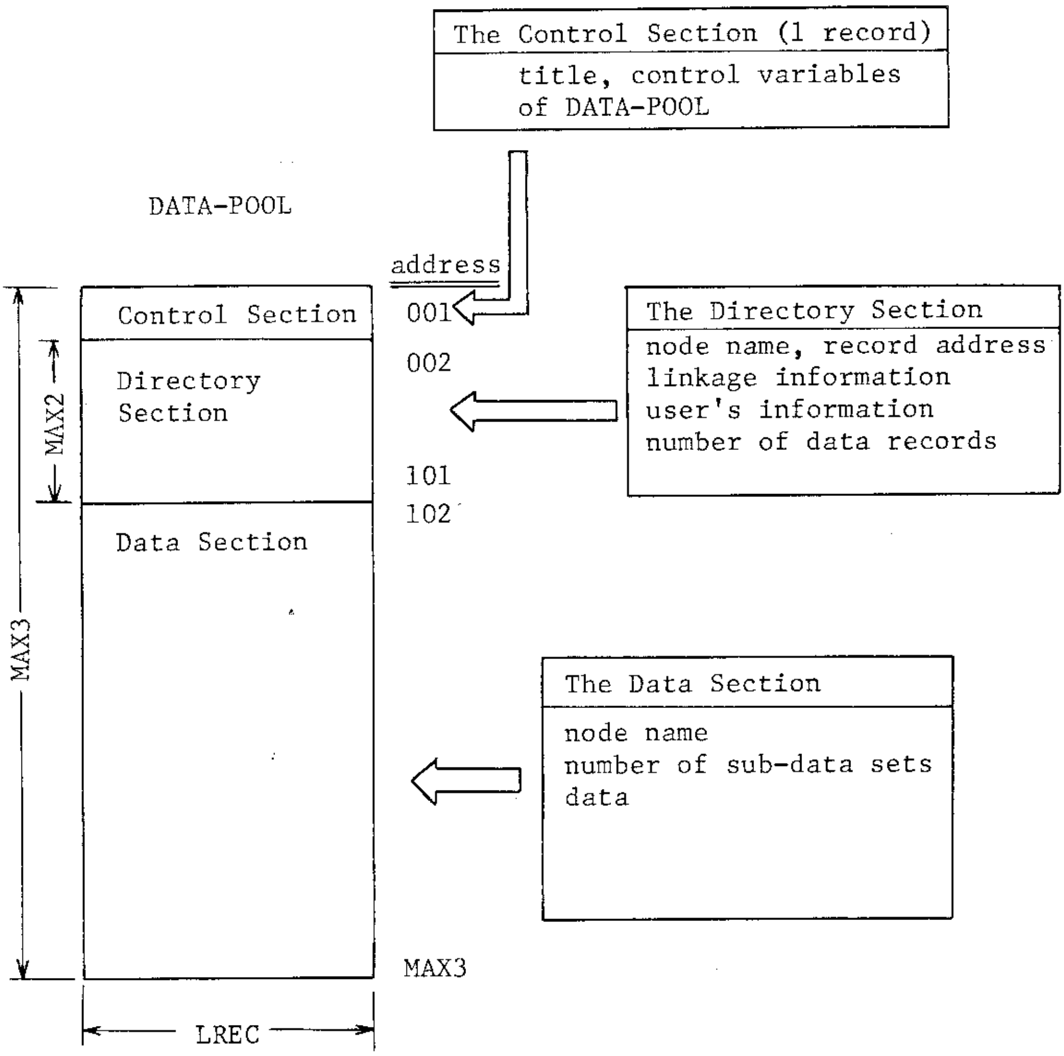


Fig. 2.1 Basic concept of DATA-POOL

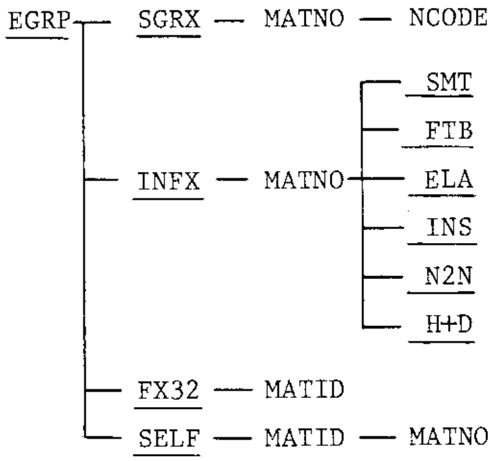


Fig. 2.2 Fundamental node tree structure

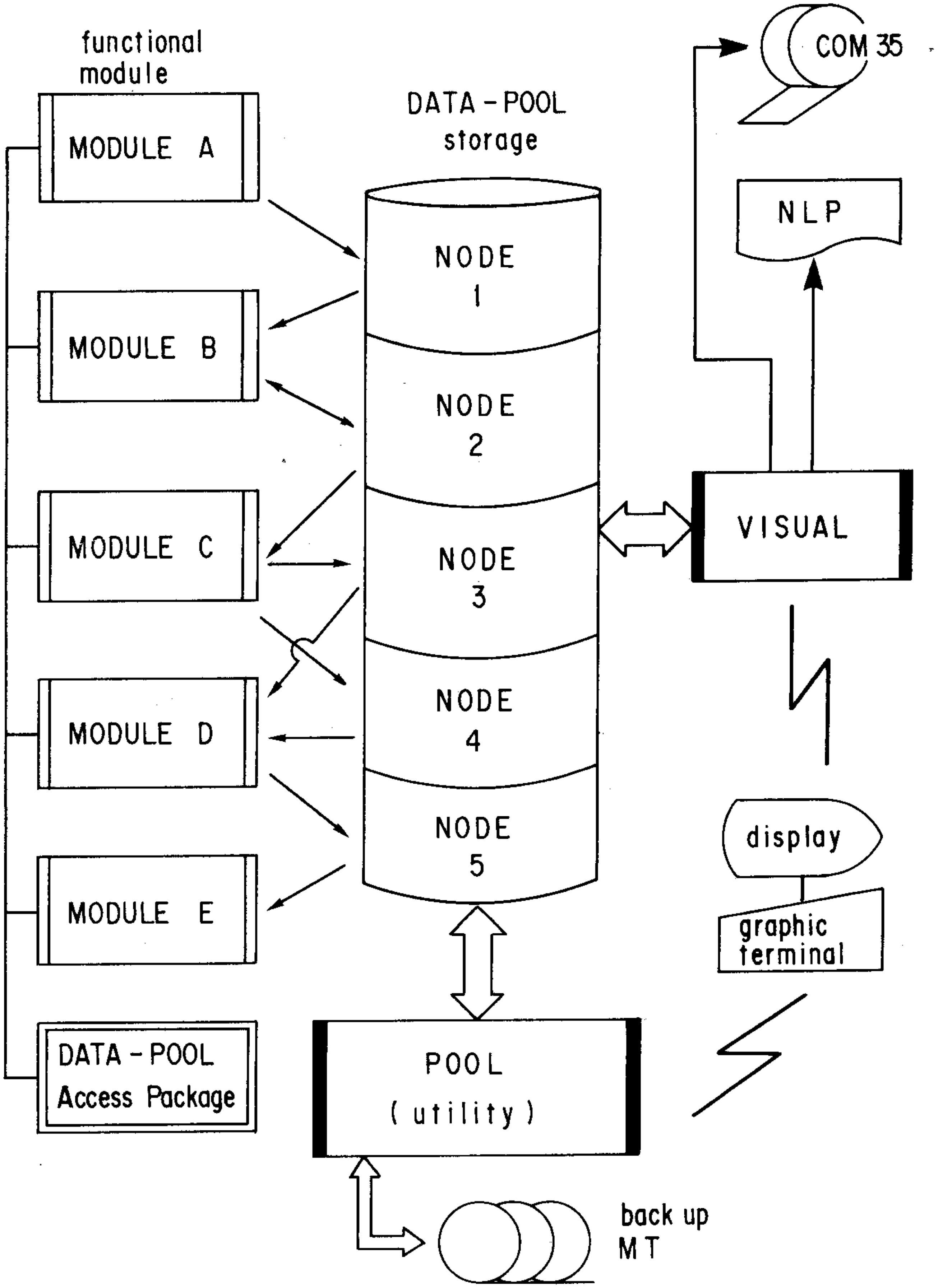


Fig. 2.3 DATA-POOL system structure

No.	Variable	Data Information
1	TITLE(1)	title of the DATA-POOL data set name, revised data, contents of the DATA-POOL et al.
~	~	
20	TITLE(20)	
21	NA1	address for the directory of the first level node
22	NA2	head address of the vacant directory section
23	NA3	head address of the vacant data section
24	KEY1	write flag for the exclusive control
25	KEY2	read flag for the exclusive control (not used)
26	LREC	length of a physical record (words)
27	MAX1	maximum number of the same level node
28	MAX2	size of the directory section
29	MAX3	size of the data section
30	NREAL1	number of used records in the directory section
31	NREAL2	number of used records in the data section
32	--	for future use
~		~
40	--	for future use
		dummy (not used)
LREC	--	

Fig. 2.4 Structure of the Control Section

	No.	Variable	Data Information
	1	NODE	node name
	2	DUMMY	for future use
	3	NAUP	address of the upper node directory
	4	ITEM	number of the sub-directory
sub-directory 1	5	NODES	node name of the first lower node
	6	NRECS	number of physical records
	7	NADWN	address for the directory of the lower node (zero means not exist)
	8	NADAT	address for the data set associated with this node (zero means not exist)
	9	NDASET	number of the sub-data set (zero means not exist)
	10	NDATE(1)	date of creation (YY-MM-DD)
	11	NDATE(2)	YY:year, MM:month, DD:day
	12	INFOM(1)	information defined by the user
	13	INFOM(2)	information defined by the user
	14	INFOM(3)	information defined by the user
	15	INFOM(4)	information defined by the user
	16	INFOM(5)	information defined by the user
sub-directory 2	17	NODE	node name
	18		
	19		
	.		
	.		
	.		
	.		
	.		
	.		
	.		
	LREC		

Fig. 2.5 Structure of the Directory Section

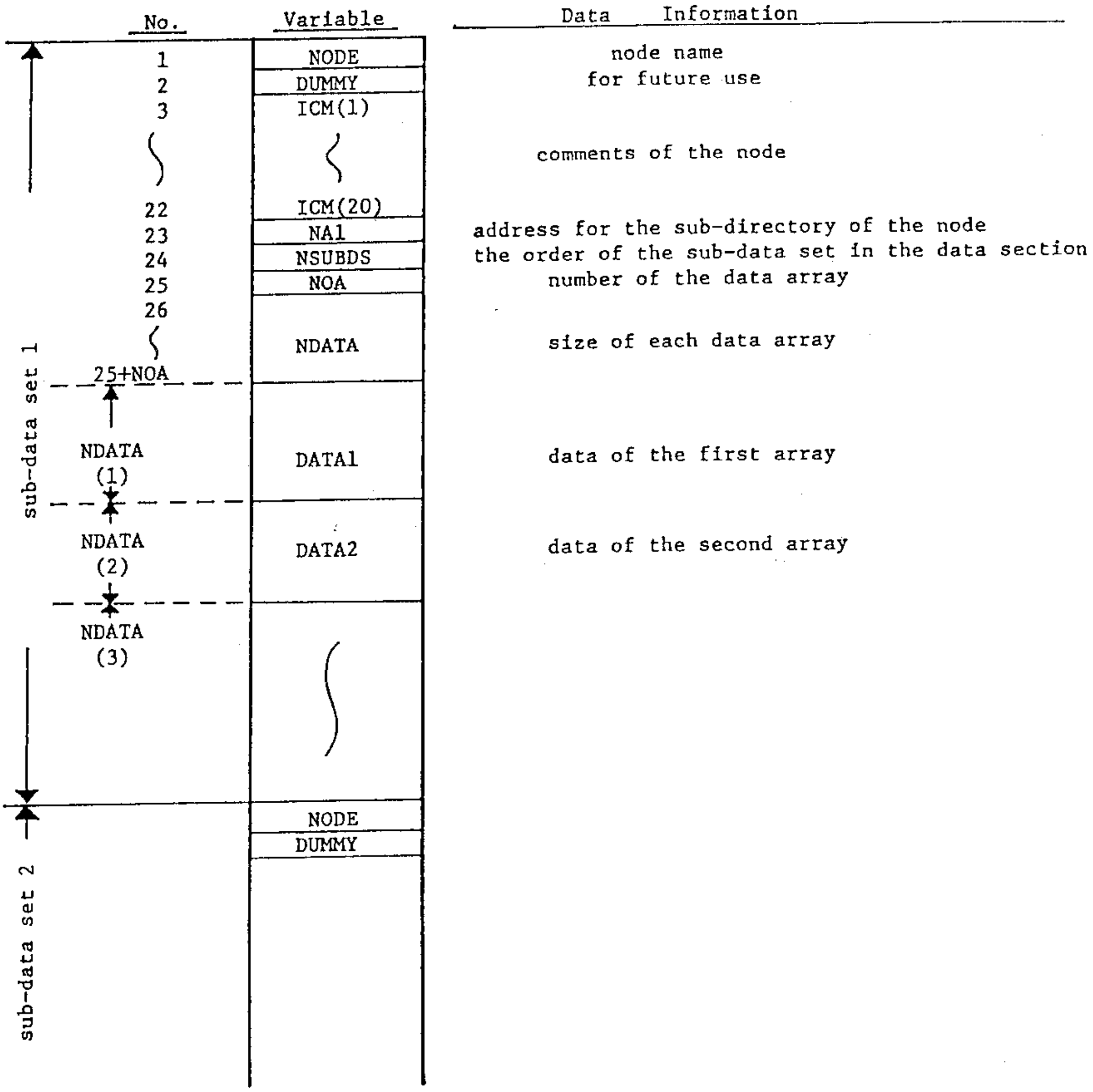


Fig. 2.6 Structure of the Data Section

3. Function of DATA-POOL

The access for the DATA-POOL is carried out by using access subroutines. Users can easily treat the data in the DATA-POOL to call the access subroutines in user's program written in FORTRAN77 language. An outline of the access subroutines is shown in Section 3.1 and the description of the common table is shown in Section 3.2. The variables of each access subroutine are described in Section 3.3.

3.1 Access Subroutine

Access subroutines are written in FORTRAN77 language. A subroutine GETDCB is only written in ASSEMBLER language. 29 access subroutines are stored in the DATA-POOL access package and these functions are as follows:

(1) PINIT : initialize DATA-POOL and clear the Control Section,

(2) POPEN : declare the access of DATA-POOL and open the data set,

(3) PWSTAT: declare the start of writing to DATA-POOL and set the exclusive control,

(4) PWEND : declare the end of writing to DATA-POOL and reset the exclusive control,

(5) PSET : set the node name and record address to the Directory Section in order to write the data,

(6) PFIND : retrieve the node name and record address from the Directory Section in order to read the data,

(7) PRITE : write data to the Data Section,
~
PRITE4

(8) PREAD : read data from the Data Section,
~
PREAD4

(9) PDELT : delete node name and data,

(10) PAGET : retrieve next record address to be read or write access,

(11) PDGET : retrieve the directory information,

(12) PASTO : set record address to be read or write access,

(13) PSKIP : skip arbitrary logical records,

(14) SETMSG: set the maximum record size of DATA-POOL,

(15) WRTCHK: set the number of records to be write,

(16) NODEER: error check,

(17) CATLST: display the Directory Section,

(18) GETDCB: retrieve the DCB information of the data set.

The other subroutines are supplementary ones for the subroutines described above, so that the descriptions are abbreviated.

3.2 Common Table

The control information of DATA-POOL is set to the common area, and used by each subroutine. The initialization of the variable is carried out by using a subroutine PINIT or POPEN. The variables in five common tables are described below.

(a) /DPCONT/

LCONTR : size of control section (normally 40 is set),

NCONTR : maximum number of allocated files (99 is set),

ICONTR(40,99) : informations of the Control Section,

IX : address variable of the direct access file,

NSUBDS : number of sub-data set to be write in a data set,

NDSTAT : start address to be write.

(b) /DPWORK/

LBUFFR : size of buffer area (1000) greater than a physical record length of DATA-POOL,

LRECOD : length of a physical record,

IBUFFR(1000) : working area for input/output access,

NRECOD : maximum number of physical records,

NODE1
NODE2
NADWN
NADAT
NDASET
NDATE(2)
NINFOM(5) } information of the sub-directory,

NUTOLD : logical unit number recently accessed,

NTHOLD : level of node recently accessed, NODOLD(10,2) : node name and address of sub-directory recently accessed,

NA1 : address of directory.

(c) /DPWCHK/

NRPEMT : number of records possible to write,

NRWRTN : number of physical records written in the Data Section,

NIXOLD : starting record address to be write,

NUTWTN : logical unit number to be write,

NTHWTN : node level to be write,

NODWTN(10,3) : history of directory and sub-directory for the node,

(1,1) ; node name

(1,2) ; history flag for sub-directory

(1,3) ; history flag for directory

flag : 0 = newly created

1 = not used

2 = no update

3 = address of lower directory in the sub-directory is updated

4 = address of data set in the sub-directory is updated

ISDBEF(12) : backup area of old sub-directory when the sub-directory is updated.

(d) /DPDELT/

IBUFF2(1000) : working area should be the same size of IBUFFR.

(e) /DPEMSG/

IFLAG : flag whether an error of error No. 232 occurs or not,

NERNO : error No.

3.3 PINIT Subroutine

PINIT initializes a direct-access data set of the DATA-POOL and sets the maximum number of physical records.

calling sequence : PINIT (NUNIT, NDIRCT, LENGTH, ITITLE)

[Input]   NUNIT : logical unit number (1 ~ 99),
          NDIRCT : number of directory records,
          LENGTH : physical record length (words),
          ITITLE : title of DATA-POOL (80 words).

subroutines called : SETMSG, ERRSET, DATE

3.4 POPEN Subroutine

POPEN declares the use of DATA-POOL and sets DCB information of the data set. This subroutine must be called as the first one when the access for DATA-POOL is done.

calling sequence : POPEN (NUNIT, JCONTR)

[Input]   NUNIT : logical unit number (1 ~ 99),
[Output] JCONTR : data of the Control Section (40 words).

subroutines called : GETDCB, DATE

3.5 PWSTAT Subroutine

PWSTAT declares the start of writing and sets the exclusive control.

calling sequence : PWSTAT (NUNIT)

[Input]   NUNIT : logical unit number
subroutines called : none

3.6 PWEND Subroutine

PWEND declares the end of writing and resets the exclusive control.

calling sequence : PWEND (NUNIT)

[Input]   NUNIT : logical unit number
subroutines called : none

3.7 PSET Subroutine

PSET sets the node name to the Directory Section of the DATA-POOL and the address to be write. This subroutine must be called before PRITE ~ PRITE4 statements.

calling sequence : PSET (NUNIT, NODE, NTH, INFOM, NUPDAT, NRETUN)

[Input] NUNIT : logical unit number,
NODE : node names from 1 to NTH levels to be set,
NTH : number of levels for node names to be set,
INFOM : user information to be set to the sub-directory section (5),
NUPDAT : condition flag to be update,
0 = node name and address are not set when the same node and data already exist.
1 = node name and address are reset although the same node already exists.

[Output] NRETUN : return condition,
0 = set node name and address to be write.
1 = node name and address are not set because the same node exists or abnormal operations are required.

subroutines called : DCLEAR, DATDLT

The general flow of the PSET subroutine is shown in Fig. 3.1.

3.8 PFIND Subroutine

PFIND searches an assigned node name in the Directory Section and set the record address to be read. This subroutine must be called before PREAD ~ PREAD4 statements.

calling sequence : PFIND (NUNIT, NODE, NTH, NDIRC, LLL)

[Input] NUNIT : logical unit number,

NODE : node names from 1 to NTH levels to be search,

NTH : number of levels for node names to be search.

[Output] NDIRC : data of the sub-directory of the assigned node name (12),

LLL : return condition,

0 = normal return.

800 = sub-directory not exists.

1×× = node name is strange.

2×× = address of the directory is strange.

900 = level of the node name is strange.

subroutines called: none

The general flow of the PFIND subroutine is shown in Fig. 3.2.

3.9 PRITE ~ PRITE4 subroutines

These subroutines store the data in the Data Section of DATA-POOL. PRITE stores only comments. PRITE1 ~ PRITE4 store the one-dimensional data consisted of 1 to 4 sets, respectively.

```
calling sequence : PRITE (NUNIT, ICM)
                   PRITE1(NUNIT, ICM, N1, D1)
                   PRITE2(NUNIT, ICM, N1, D1, N2, D2)
                   PRITE3(NUNIT, ICM, N1, D1, N2, D2, N3, D3)
                   PRITE4(NUNIT, ICM, N1, D1, N2, D2, N3, D3, N4, D4)
```

[Input] NUNIT : logical unit number,
ICM : comments of the data (20 words),
N1 ~ N4 : size of the arrays D1 ~ D4,
D1 ~ D4 : data arrays to be write.

subroutines called : DCLEAR, WRTCHK

3.10 PREAD ~ PREAD4 Subroutines

These subroutines read the data in the Data Section according to the record sequence written by PRITE ~ PRITE4 statements. PREAD reads comments and record information of the node name. PREAD1 ~ PREAD4 read the one-dimensional data of 1 to 4 sets corresponding to PRITE ~ PRITE4 statements, respectively.

```
calling sequence : PREAD (NUNIT, NAME1, NAME2, ICM, NASBD, NOSBDS, NOARY,
                   NDATA)
                   PREAD1(NUNIT, ICM, N1, D1)
                   PREAD2(NUNIT, ICM, N1, D1, N2, D2)
                   PREAD3(NUNIT, ICM, N1, D1, N2, D2, N3, D3)
                   PREAD4(NUNIT, ICM, N1, D1, N2, D2, N3, D3, N4, D4)
```

[Input] NUNIT : logical unit number,
NAME1 : node name of the data in the Data Section,
NAME2 : not used (for future use),
ICM : comments of the data (20 words),
NASBD : address for the sub-directory of the node name,
NOSBDS: the order of the sub-data set in the Data Section,

NOARY : number of data arrays,
NDATA : size of each array (NOARY),
N1~N4 : size of the arrays D1~D4,
D1~D4 : data arrays to be read.

subroutines called : none

3.11 PDELT Subroutine

PDELT erases directories and data under the assigned node name from the DATA-POOL.

calling sequence : PDELT (NUNIT, NODE, NTH, NRETUN)

[Input] NUNIT : logical unit number,
NODE : node names to be erase,
NTH : number of levels for the node name.

[Output] NRETUN: return condition,
0 = normal return.
1 = sub-directory of the node name not exist.

subroutines called : none

3.12 PAGET Subroutine

PAGET gets the record address assigned when the PFIND or PSET subroutine is executed. This subroutine is ordinarily used after the PFIND statements, and the obtained record address is used by the PASTO subroutine in order to read some data in the Data Section.

calling sequence : PAGET(N)

[Output] N : direct-access record address.

subroutines called : none

3.13 PDGET Subroutine

PDGET gets information of the directory record assigned by the user.

calling sequence : PDGET (NUNIT, NODE, NTH, ITEM, NSDIRC)

[Input]   NUNIT : logical unit number,
           NODE : node names from 1 to NTH levels,
           NTH   : number of levels for the node names.
[Output]   ITEM : number of sub-directories in the directory,
           NSDIRC: information of each sub-directory
                   (12, ITEM).

subroutines called : none

3.14 PASTO Subroutine

PASTO sets the direct-access record address to be read or write. The PREAD/PRITE access after the PASTO statement is performed from the record address assigned by the user.
calling sequence : PASTO (N)

[Input]   N   : direct-access record address.
subroutines called : none

3.15 PSKIP Subroutine

PSKIP skips over some records assigned by the user.
calling sequence : PSKIP (NUNIT, N)

[Input]   NUNIT : logical unit number,
           N   : number of logical record to be skip.
                   (An execution of PRITE statement corresponds to a logical record.)
subroutines called : none

3.16 SETMSG Subroutine

SETMSG stores a FORTRAN error No. 232 to the variable NEMSG.
calling sequence : SETMSG (RET, ERRNO, N1, N2)
subroutine called : none

3.17 WRTCHK Subroutine

WRTCHK sets numbers of physical and logical records of data written in the DATA-POOL to the Control and the Directory Sections.

calling sequence : WRTCHK (NUNIT, IXOLD)

[Input] NUNIT : logical unit number,
IXOLD : direct-access record written in the Data Section.

subroutines called : none

3.18 NODEER Subroutine

NODEER is an error routine to print error message.

calling sequence : NODEER (NUNIT, NTH, NODE)

[Input] NUNIT : logical unit number,
NTH : number of levels for node names,
NODE : node names from 1 to NTH levels.

subroutines called : none

3.19 CATLST Subroutine

CATLST prints the Control and the Directory Sections of the DATA-POOL in order to obtain record information.

calling sequence : CATLST (NUNIT)

[Input] NUNIT : logical unit number.

3.20 GETDCB Subroutine

GETDCB is written in ASSEMBLER language and obtains DCB information of the DATA-POOL.

calling sequence : GETDCB (DDNAME, LRECL, LBLKS, RECFM, DSORG, IR)

[Input] DDNAME : DD name (FT91F001 etc.)[8 bytes character],

[Output] LRECL : record length of the direct-access data set [4 bytes integer],

LBLKS : block size of the direct-access data
set [4 bytes integer],

RECFM : record format of the direct-access data
set [4 bytes character],

DSORG : data set organization [4 bytes character],

IR : return condition,
0 = normal return.
8 = DD name not exists.

subroutines called : none

3.21 SUBDLT Subroutine

SUBDLT erases directories and data stored in the sub-directory.

calling sequence : SUBDLT (NUNIT, NSDOLD)

[Input] NUNIT : logical unit number,
NSDOLD : sub-directory to be erase (12).

subroutines called : DIRDLT, DATDLT

3.22 DIRDLT Subroutine

DIRDLT erases the directory assigned in the sub-directory.
The erase is carried out by setting '////' to the variable of node name.

calling sequence : DIRDLT (NUNIT, NSDOLD)

[Input] NUNIT : logical unit number,
NSDOLD: sub-directory (12).

subroutines called : none

3.23 DATDLT Subroutine

DATDLT erases the data assigned in the sub-directory.
The erase is carried out by setting '////' to the variable of node name.

calling sequence : DATDLT (NUNIT, NSDOLD)

[Input]   NUNIT : logical unit number,
           NSDOLD: sub-directory (12).
subroutines called : none

3.24 DCLEAR Subroutine

DCLEAR recovers the directory when record overflow is occured in PSET or PRITE subroutines.

calling sequence : DCLEAR
subroutines called : none

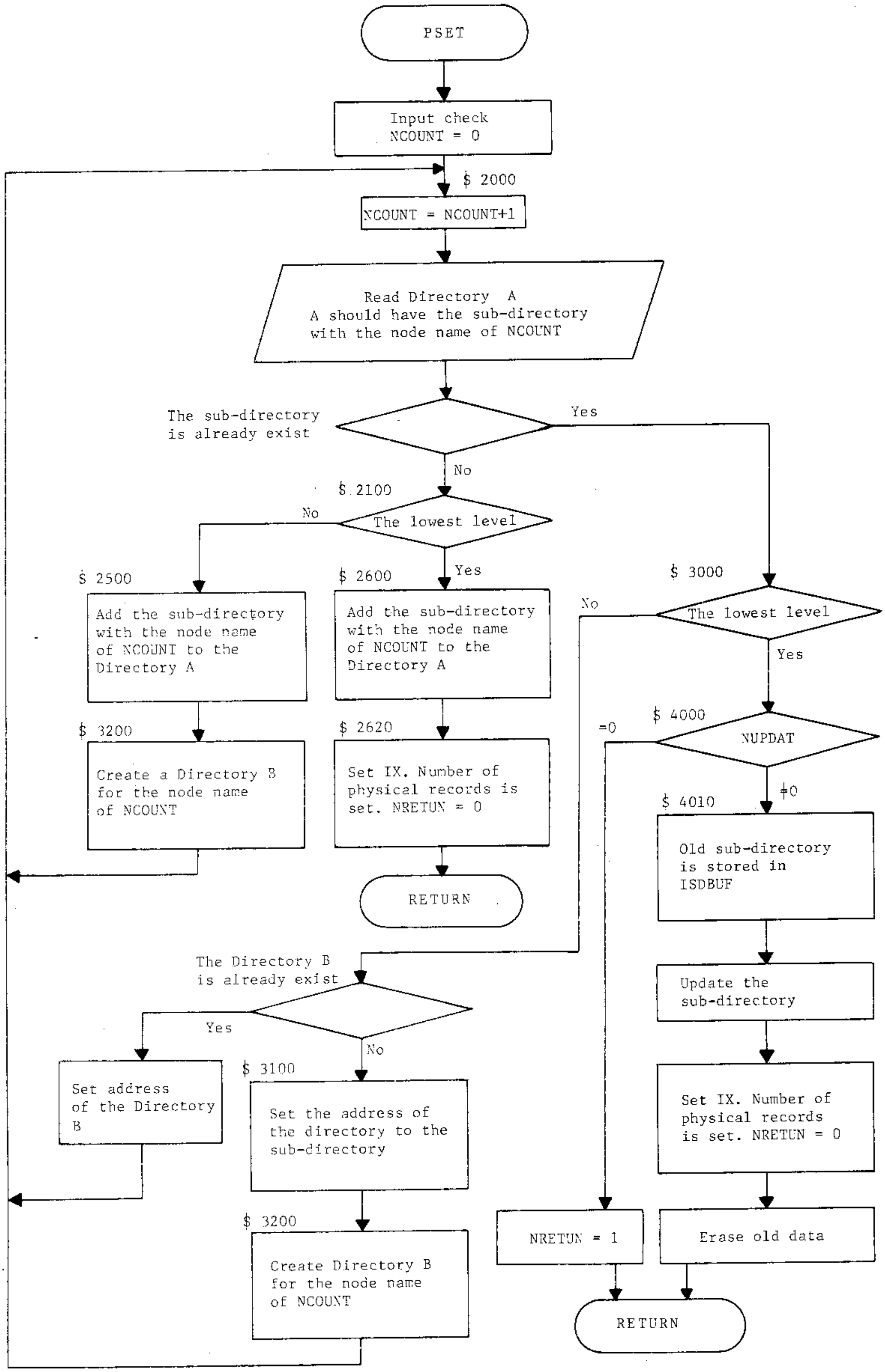


Fig. 3.1 Flow chart of PSET subroutine

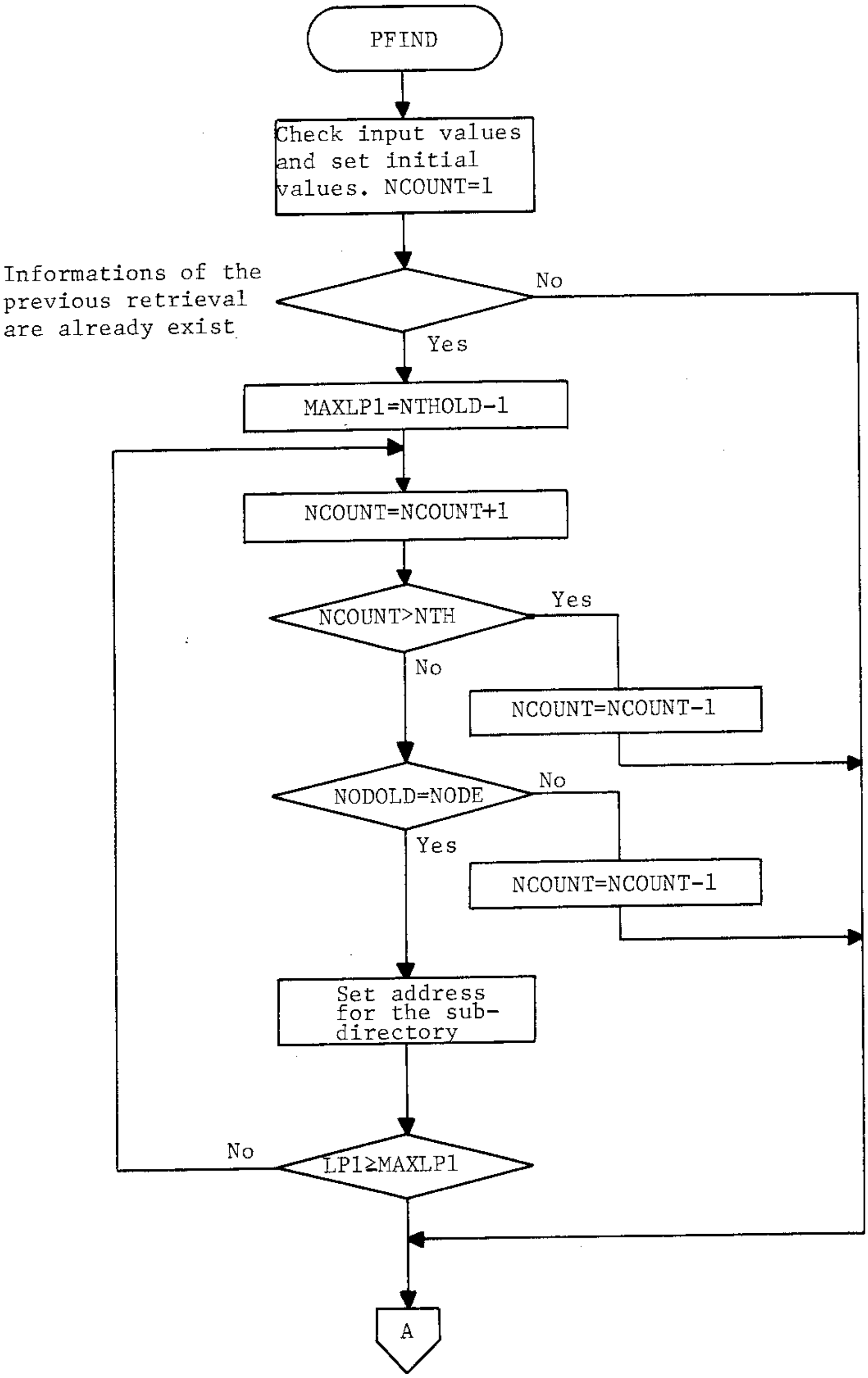

PFIND
Check input values and set initial values. NCOUNT=1
Informations of the previous retrieval are already exist
No
Yes
MAXLP1=NTHOLD-1
NCOUNT=NCOUNT+1
NCOUNT>NTH
Yes
No
NCOUNT=NCOUNT-1
NODOLD=NODE
No
Yes
NCOUNT=NCOUNT-1
Set address for the sub-directory
No
LP1≧MAXLP1
A

Fig. 3.2 Flow chart of PFIND subroutine

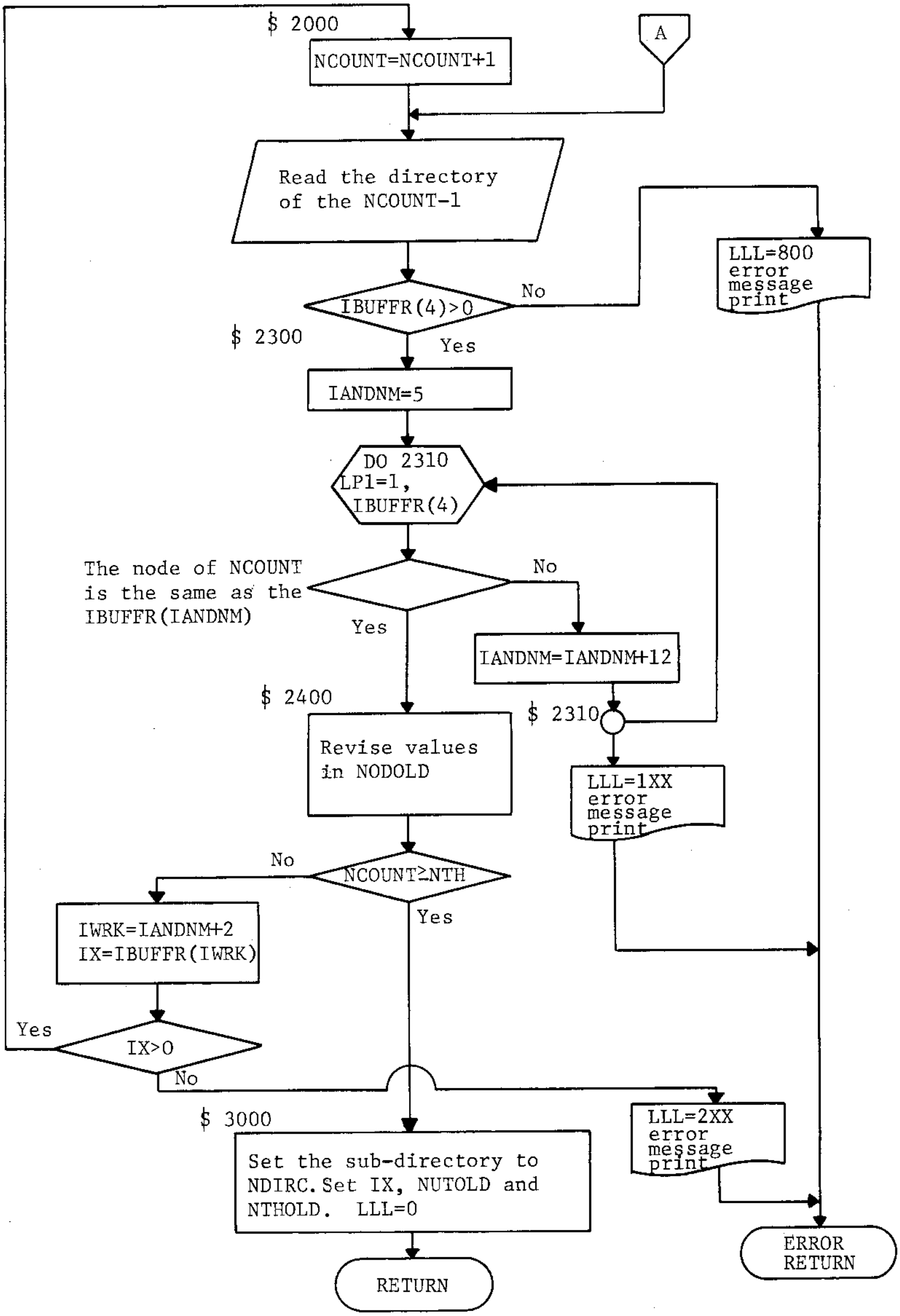

$ 2000
NCOUNT=NCOUNT+1
A
Read the directory of the NCOUNT-1
IBUFFR(4)>0
No
LLL=800 error message print
$ 2300
Yes
IANDNM=5
DO 2310 LP1=1, IBUFFR(4)
The node of NCOUNT is the same as the IBUFFR(IANDNM)
No
Yes
IANDNM=IANDNM+12
$ 2400
$ 2310
Revise values in NODOLD
LLL=1XX error message print
No
NCOUNT≥NTH
Yes
IWRK=IANDNM+2
IX=IBUFFR(IWRK)
Yes
IX>0
No
$ 3000
LLL=2XX error message print
Set the sub-directory to NDIRC. Set IX, NUTOLD and NTHOLD. LLL=0
ERROR RETURN
RETURN

Fig. 3.2 (continued)

4. Access Method for DATA-POOL

This section describes the access method of the DATA-POOL by using the access subroutines noted in the previous Section. The DATA-POOL is a direct-access data set, so that an initialization must be performed when the data set is allocated. The initialization is carried out by using INIT command in a TSS Management utility POOL described in the next Section. The allocation of the data set with DD statement is as follows :

```
//FT01F001   DD   DSN=J3679.DATAPOOL.DATA, UNIT=D0954,
// SPACE=(TRK,(50,10)),  DCB=(LRECL=3600, BLKSIZE=3600, RECFM=F),
// LABEL=(,,, OUT)
```

or

```
// EXPAND DISKTO, DDN=FT01F001, DSN=J3679.DATAPOOL.DATA,
// MODE=OUT
```

or on TSS terminal

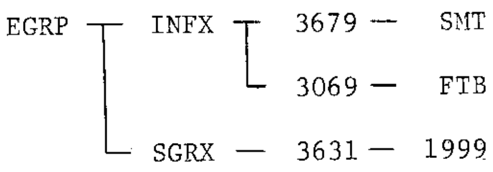
```
ATTR DCB LR(3600) BL(3600) REC(F)
ALLOC DA (DATAPOOL.DATA) UNIT(D0954) SP(50 10) T US(DCB) CAT
```

The initialized data set can be used as a DATA-POOL. A POPEN statement must be called in the user's program in order to access the DATA-POOL before the other access statements appear.

4.1 Generation of Node Structure

The sequence generating node structure is described by using a sample as follows :

A sample node structure is



The generation of the node structure is carried out by using PSET, PWSTAT, PWEND and POPEN statements.

```
CHARACTER  JCONTR(40), NODE(10), INFOM(5)
CALL POPEN (1, JCONTR)
CALL PWSTAT(1)
```

```
      NTH=1
      NODE(1)='EGRP'
      CALL PSET(1, NODE, NTH, INFOM, 0, L)
C---DATA WITH 'EGRP' CAN BE WRITTEN AT THE POSITION
      NTH=2
      NODE(2)='INFX'
      CALL PSET(1, NODE, NTH, INFOM, 0, L)
C---DATA WITH 'INFX' CAN BE WRITTEN AT THE POSITION
      NTH=3
      NODE(3)='3679'
      CALL PSET(1, NODE, NTH, INFOM, 0, L)
C---DATA WITH '3679' CAN BE WRITTEN AT THE POSITION
      NODE(3)='3069'
      CALL PSET(1, NODE, NTH, INFOM, 0, L)
C---DATA WITH '3069' CAN BE WRITTEN AT THE POSITION
      NODE(3)='3679'
      NTH=4
      NODE(4)=' SMT'
      CALL PSET(1, NODE, NTH, INFOM, 0, L)
C---DATA WITH ' SMT' CAN BE WRITTEN AT THE POSITION
      NODE(3)='3069'
      NODE(4)=' FTB'
      CALL PSET(1, NODE, NTH, INFOM, 0, L)
C---DATA WITH ' FTB' CAN BE WRITTEN AT THE POSITION
      NODE(2)='SGRX'
      NTH=2
      CALL PSET(1, NODE, NTH, INFOR, 0, L)
      NODE(3)='3631'
      NTH=3
      CALL PSET(1, NODE, NTH, INFORM, 0, L)
C---DATA WITH '3631' CAN BE WRITTEN AT THE POSITION
      NTH=4
      NODE(4)='1999'
      CALL PSET(1, NODE, NTH, INFOM, 0, L)
C---DATA WITH '1999' CAN BE WRITTEN AT THE POSITION
      CALL PWEND(1)
```

If a node name assigned by the user already exist in the DATA-POOL, the variable L in the PSET subroutine is set to 1, and the registration of the node name is not executed by the condition of NUPDAT=0. In the period executing between PWSTAT and PWEND statements, the other write accesses to the DATA-POOL is inhibited. The user information of 5 words for each node can be stored by using the PSET statement to set the variables of INFOM. Arbitrary tree structures can be generated by the user as shown in the sample problem.

4.2 Storage Procedure

PRITE, PRITE1, PRITE2, PRITE3 and PRITE4 statements are prepared to store data in the Data Section of the DATA-POOL.

These statements must be located after the PSET statement. The combination of PRITE statements is arbitrarily defined by the user. A sample is shown as follows :

```
      CHARACTER JCONTR(40), NODE(10), ICM(20)
      DIMENSION A(100), B(200), C(300), D(400), INFOM(5)
      IA=100
      IB=200
      IC=300
      ID=400
      CALL POPEN(1, JCONTR)
      CALL PWSTAT(1)
      NTH=1
      NODE(1)='EJAE'
      CALL PSET(1, NODE, NTH, INFOM, 0, L)
      IF(L.NE.0) GO TO 7
      CALL PRITE1(1, ICM, IA, A)
    7 NTH=2
      NODE(2)='3679'
      CALL PSET(1, NODE, NTH, INFOM, 0, L)
      IF(L.NE.0) GO TO 77
      CALL PRITE(1, ICM)
   77 NTH=3
      NODE(3)='GOOD'
      CALL PSET(1, NODE, NTH, INFOM, 0, L)
      IF(L.NE.0) GO TO 777
      CALL PRITE2(1, ICM, IB, B, IC, C)
      CALL PRITE4(1, ICM, IA, A, IB, B, IC, C, ID, D)
      CALL PRITE3(1, ICM, IA, A, IB, B, IC, C)
  777 CALL PWEND(1)
```

The above sample generates the following tree structure and stores data of arrays A, B, C and D.

Tree structure	EJAE —	3679 —	GOOD
stored data	A	none	B, C
arrays			A, B, C, D
			A, B, C

The comments for data can be stored by setting the variable ICM before PRITE ~ PRITE4 statements. Many kinds of data can be stored with a node name. The record sequence can be arbitrarily defined by the user.

4.3 Retrieval Procedure

Node names and data stored in the DATA-POOL are retrieved by using PFIND, PREAD, PREAD1, PREAD2, PREAD3 and PREAD4 statements. PDGET, PAGET, PASTO and PSKIP statements may be also used to retrieve the data skillfully. A sample procedure is shown for the data stored in the previous Section.

```
CHARACTER JCONTR(40), NODE(10), ICM(20)
DIMENSION A(100), B(200), C(300), D(400), INFOM(5), NDIRC(12)
DIMENSION NDATA(4)
IA=0
IB=0
IC=0
ID=0
NTH=1
NODE(1)='EJAE'
CALL PFIND(1, NODE, NTH, NDIRC, L)
IF(L.NE.0) GO TO 999
INFOM(1)=NDIRC(8)
INFOM(2)=NDIRC(9)
INFOM(3)=NDIRC(10)
INFOM(4)=NDIRC(11)
INFOM(5)=NDIRC(12)
CALL PREAD1(1, ICM, IA, A)
NTH=2
NODE(2)='3679'
```

```
      CALL PFIND(1, NODE, NTH, NDIRC, L)
      IF(L.NE.0) GO TO 999
      DO 10 I=1, 5
      K=I + 7
10    INFOM(I)=NDIRC(K)
      CALL PREAD(1, N1, N2, ICM, N3, N4, N5, NDATA)
      NTH=3
      NODE(3)='GOOD'
      CALL PFIND(1, NODE, NTH, NDIRC, L)
      IF(L.NE.0) GO TO 999
      DO 11 I=1, 5
      K=I + 7
11    INFOM(I)=NDIRC(K)
      CALL PREAD2(1, ICM, IB, B, IC, C)
      CALL PREAD4(1, ICM, IA, A, IB, B, IC, C, ID, D)
      CALL PREAD3(1, ICM, IA, A, IB, B, IC, C)
      :
      :
C---ERROR MESSAGE DISPLAY
999   CALL NODEER(1, NTH, NODE)
```

In the sequence of the sample procedure, the variables IA, IB, IC and ID are initially set zero. However, the variables are set the sizes of data arrays after PREAD statements, so that the initial setting may not be necessary. Information defined by the user are stored in NDIRC(8) ~ NDIRC (12) arrays after PREAD statements. An error monitor will be located at the end of the user's program in order to detect errors caused by mistaken conditions.

If the user wish to read only some parts of data, PSKIP statement may be used as follows :

```
      NTH=3
      NODE(3)='GOOD"
      CALL PFIND(1, NODE, NTH, NDIRC, L)
      IF(L.NE.0) GO TO 999
      CALL PREAD2(1, ICM, IB, B, IC, C)
      CALL PSKIP(1, 1)
      CALL PREAD3(1, ICM, IA, A, IB, B, IC, C)
      :
      :
```

PAGET and PASTO statements may be used to read data in a same category for each node name as follows :

```
          DO 100 N=1, 10
          CALL PSET(1, NODE(1, N), NTH, INFOM, 0, L)
          DO 100 M=1, 20
   100    CALL PRITE1(1, ICM, N1, A(1, M))
          :
          :
          DO 200 N=1, 10
          CALL PFIND(1, NODE(1, N), NTH, NDIRC, L)
   200    CALL PAGET(NADRS(N))
          :
          :
          DO 300 M=1, 20
          DO 300 N=1, 10
          CALL PASTO(NADRS(N))
          CALL PREAD1(1, ICM, N1, B)
          CALL PAGET (NADRS(N))
          :
          :
   300    CONTINUE
```

A sample program to retrieve information of nuclei and atomic number densities for each material in the macroscopic cross-section library is shown in Fig. 4.1. A Job Control Language and the input data are also shown in this figure.

4.4 Limitations and Notes for Operation

a) Limitations

(i) Logical unit number is allowed from FT01F001 to FT99F001.

(ii) The maximum level for node names is 10.

(iii) Physical record length in the DATA-POOL is allowed up to 1000 words.+

(iv) DATA-POOL is a direct-access data base so that a direct-access device is essentially required.

(v) Access subroutines for DATA-POOL are written in FORTRAN77 language so that the FORTRAN77 compiler is essentially needed.

b) Notes for Operation

(i) Initialization of the DATA-POOL is executed for the area allocated with DD statement. The number of initialized records is printed in PINIT subroutine, so that user should not access over the limit.

(ii) In the period executing between PWSTAT and PWEND statements, the other write access is inhibited. Namely the other jobs may be terminated.

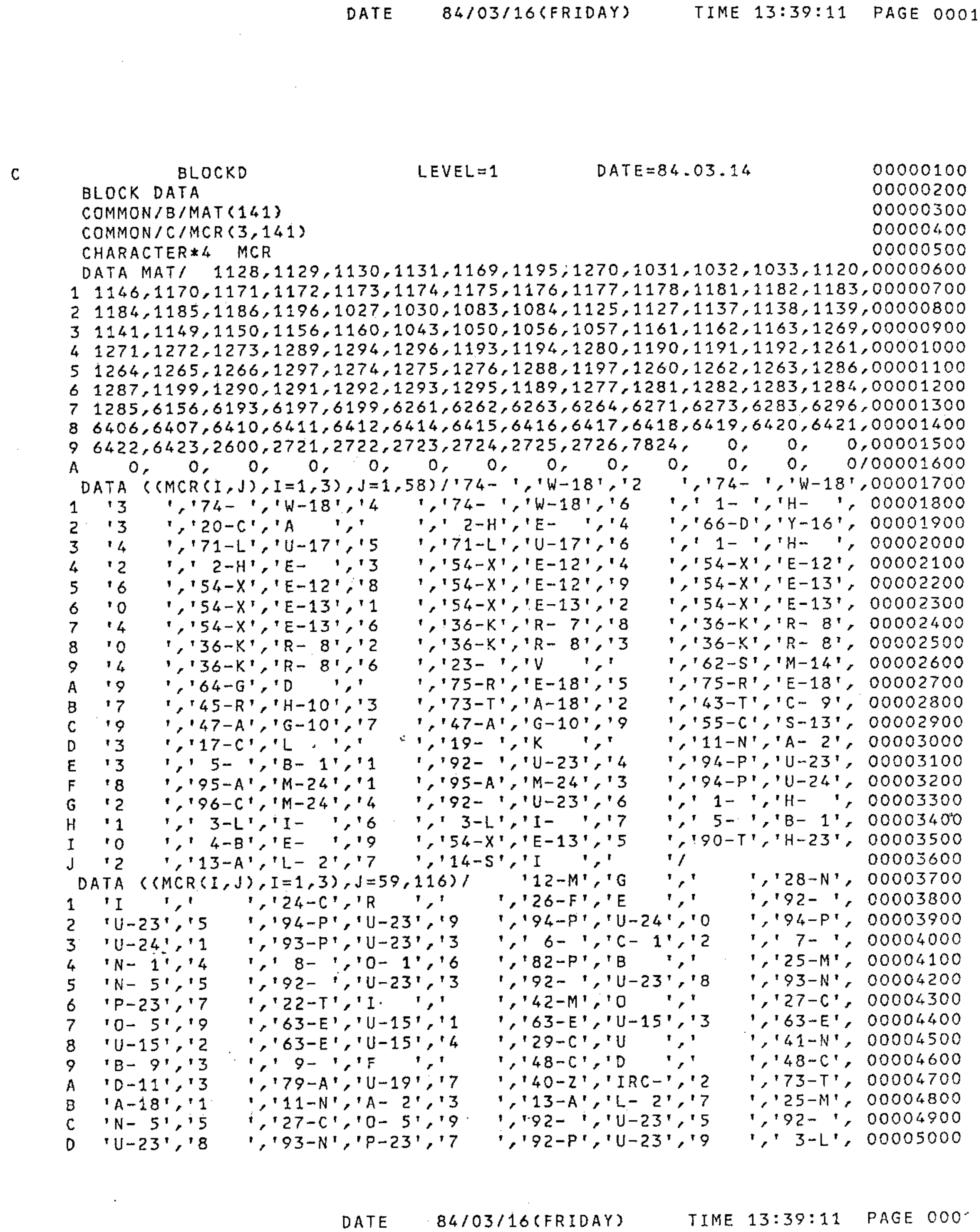
(iii) When the job is abnormally terminated, the DATA-POOL may not be generated correctly. The restoration can be performed by using following procedures.

- write flag is already on. → execute FLAG command of the TSS management utility POOL.
- node name is registered, but data with the node name are not stored. → execute DELETE command of the TSS management utility POOL.
- writing data exceeds the limit of the initialized records or the other destractions. → execute MEND command of the TSS management utility POOL.

The description of the TSS management utility POOL is shown in the next Section.

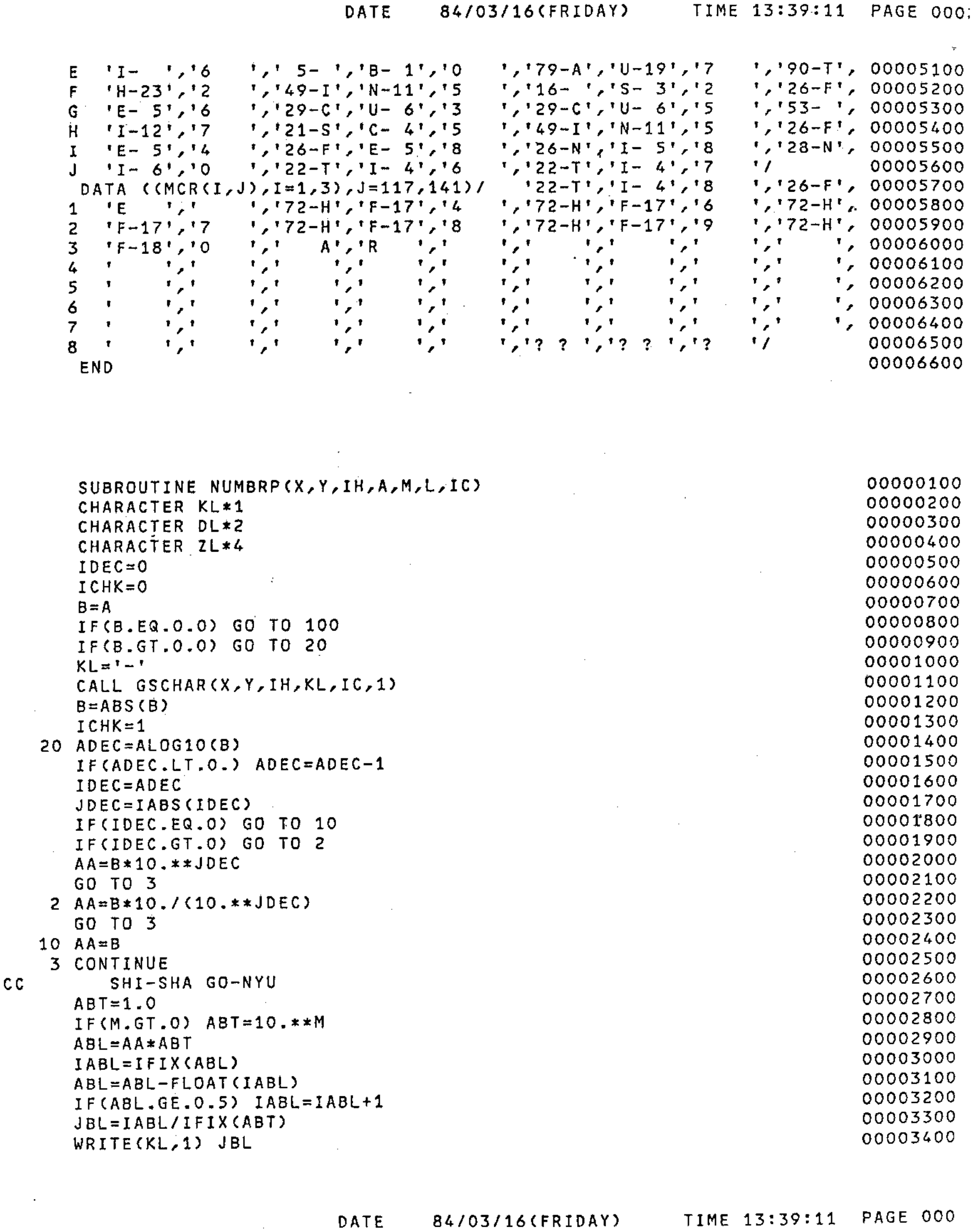
+ : The value can be changed by modifications of common blocks, PINIT and POPEN subroutines.

(iv) The node name consists of 4-characters. The characters A~Z, 0~9 and blank are allowed. For example, node names _SMT and SMT_ are different from each other.



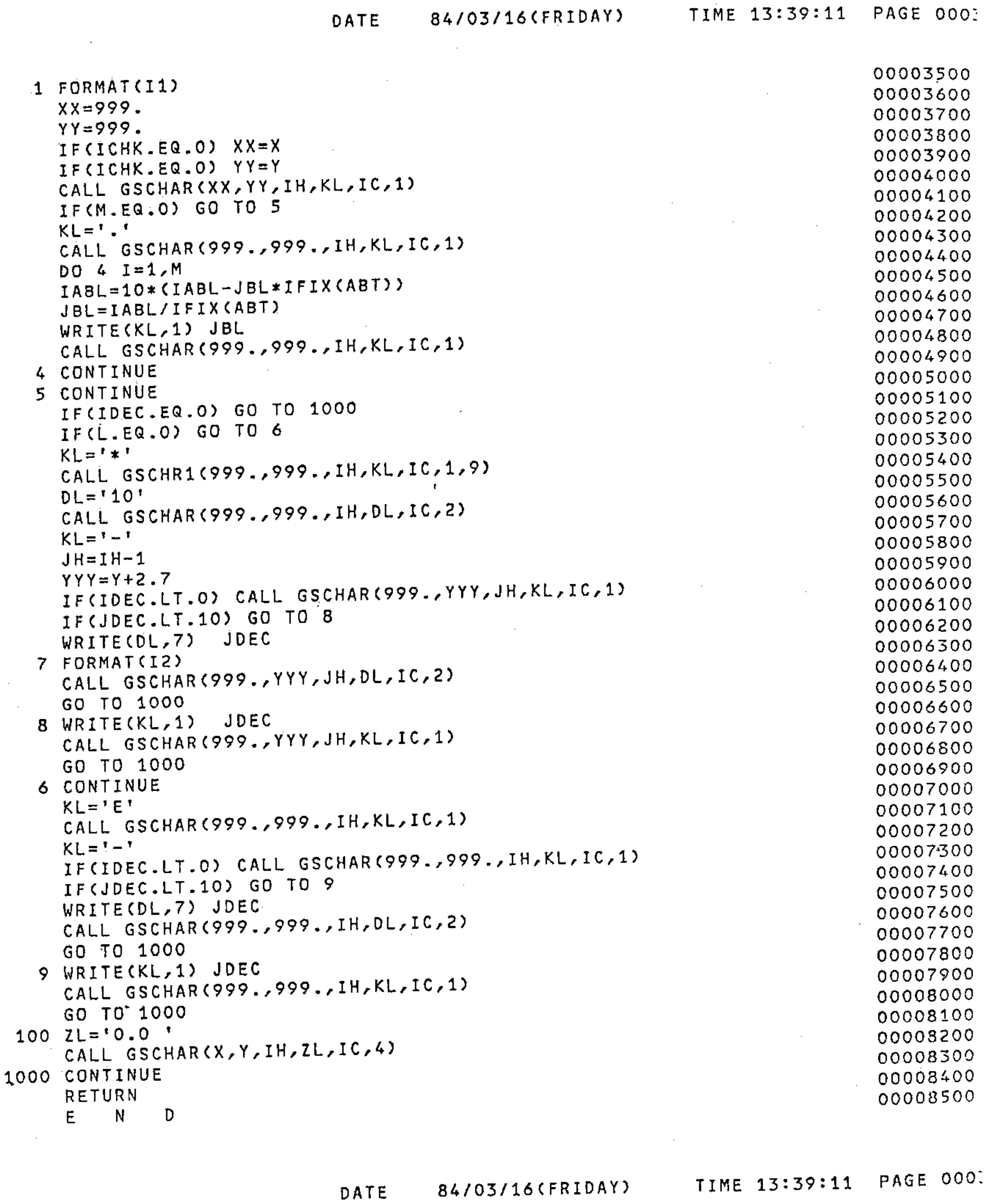
```
                          DATE    84/03/16(FRIDAY)     TIME 13:39:11  PAGE 0001



C          BLOCKD                 LEVEL=1          DATE=84.03.14           00000100
      BLOCK DATA                                                           00000200
      COMMON/B/MAT(141)                                                    00000300
      COMMON/C/MCR(3,141)                                                  00000400
      CHARACTER*4  MCR                                                     00000500
      DATA MAT/  1128,1129,1130,1131,1169,1195,1270,1031,1032,1033,1120,00000600
     1 1146,1170,1171,1172,1173,1174,1175,1176,1177,1178,1181,1182,1183,00000700
     2 1184,1185,1186,1196,1027,1030,1083,1084,1125,1127,1137,1138,1139,00000800
     3 1141,1149,1150,1156,1160,1043,1050,1056,1057,1161,1162,1163,1269,00000900
     4 1271,1272,1273,1289,1294,1296,1193,1194,1280,1190,1191,1192,1261,00001000
     5 1264,1265,1266,1297,1274,1275,1276,1288,1197,1260,1262,1263,1286,00001100
     6 1287,1199,1290,1291,1292,1293,1295,1189,1277,1281,1282,1283,1284,00001200
     7 1285,6156,6193,6197,6199,6261,6262,6263,6264,6271,6273,6283,6296,00001300
     8 6406,6407,6410,6411,6412,6414,6415,6416,6417,6418,6419,6420,6421,00001400
     9 6422,6423,2600,2721,2722,2723,2724,2725,2726,7824,   0,   0,   0,00001500
     A    0,   0,   0,   0,   0,   0,   0,   0,   0,   0,   0,   0,   0/00001600
      DATA ((MCR(I,J),I=1,3),J=1,58)/'74- ','W-18','2   ','74- ','W-18',00001700
     1  '3   ','74- ','W-18','4   ','74- ','W-18','6   ',' 1- ','H-  ', 00001800
     2  '3   ','20-C','A   ','    ',' 2-H','E-  ','4   ','66-D','Y-16', 00001900
     3  '4   ','71-L','U-17','5   ','71-L','U-17','6   ',' 1- ','H-  ', 00002000
     4  '2   ',' 2-H','E-  ','3   ','54-X','E-12','4   ','54-X','E-12', 00002100
     5  '6   ','54-X','E-12','8   ','54-X','E-12','9   ','54-X','E-13', 00002200
     6  '0   ','54-X','E-13','1   ','54-X','E-13','2   ','54-X','E-13', 00002300
     7  '4   ','54-X','E-13','6   ','36-K','R- 7','8   ','36-K','R- 8', 00002400
     8  '0   ','36-K','R- 8','2   ','36-K','R- 8','3   ','36-K','R- 8', 00002500
     9  '4   ','36-K','R- 8','6   ','23- ','V   ','    ','62-S','M-14', 00002600
     A  '9   ','64-G','D   ','    ','75-R','E-18','5   ','75-R','E-18', 00002700
     B  '7   ','45-R','H-10','3   ','73-T','A-18','2   ','43-T','C- 9', 00002800
     C  '9   ','47-A','G-10','7   ','47-A','G-10','9   ','55-C','S-13', 00002900
     D  '3   ','17-C','L   ','    ','19- ','K   ','    ','11-N','A- 2', 00003000
     E  '3   ',' 5- ','B- 1','1   ','92- ','U-23','4   ','94-P','U-23', 00003100
     F  '8   ','95-A','M-24','1   ','95-A','M-24','3   ','94-P','U-24', 00003200
     G  '2   ','96-C','M-24','4   ','92- ','U-23','6   ',' 1- ','H-  ', 00003300
     H  '1   ',' 3-L','I-  ','6   ',' 3-L','I-  ','7   ',' 5- ','B- 1', 00003400
     I  '0   ',' 4-B','E-  ','9   ','54-X','E-13','5   ','90-T','H-23', 00003500
     J  '2   ','13-A','L- 2','7   ','14-S','I   ','    '/                 00003600
      DATA ((MCR(I,J),I=1,3),J=59,116)/    '12-M','G   ','    ','28-N', 00003700
     1  'I   ','    ','24-C','R   ','    ','26-F','E   ','    ','92- ', 00003800
     2  'U-23','5   ','94-P','U-23','9   ','94-P','U-24','0   ','94-P', 00003900
     3  'U-24','1   ','93-P','U-23','3   ',' 6- ','C- 1','2   ',' 7- ', 00004000
     4  'N- 1','4   ',' 8- ','O- 1','6   ','82-P','B   ','    ','25-M', 00004100
     5  'N- 5','5   ','92- ','U-23','3   ','92- ','U-23','8   ','93-N', 00004200
     6  'P-23','7   ','22-T','I   ','    ','42-M','O   ','    ','27-C', 00004300
     7  'O- 5','9   ','63-E','U-15','1   ','63-E','U-15','3   ','63-E', 00004400
     8  'U-15','2   ','63-E','U-15','4   ','29-C','U   ','    ','41-N', 00004500
     9  'B- 9','3   ',' 9- ','F   ','    ','48-C','D   ','    ','48-C', 00004600
     A  'D-11','3   ','79-A','U-19','7   ','40-Z','IRC-','2   ','73-T', 00004700
     B  'A-18','1   ','11-N','A- 2','3   ','13-A','L- 2','7   ','25-M', 00004800
     C  'N- 5','5   ','27-C','O- 5','9   ','92- ','U-23','5   ','92- ', 00004900
     D  'U-23','8   ','93-N','P-23','7   ','92-P','U-23','9   ',' 3-L', 00005000


                          DATE    84/03/16(FRIDAY)     TIME 13:39:11  PAGE 000
```

Fig. 4.1 Sample program for information retrieval of DATA-POOL



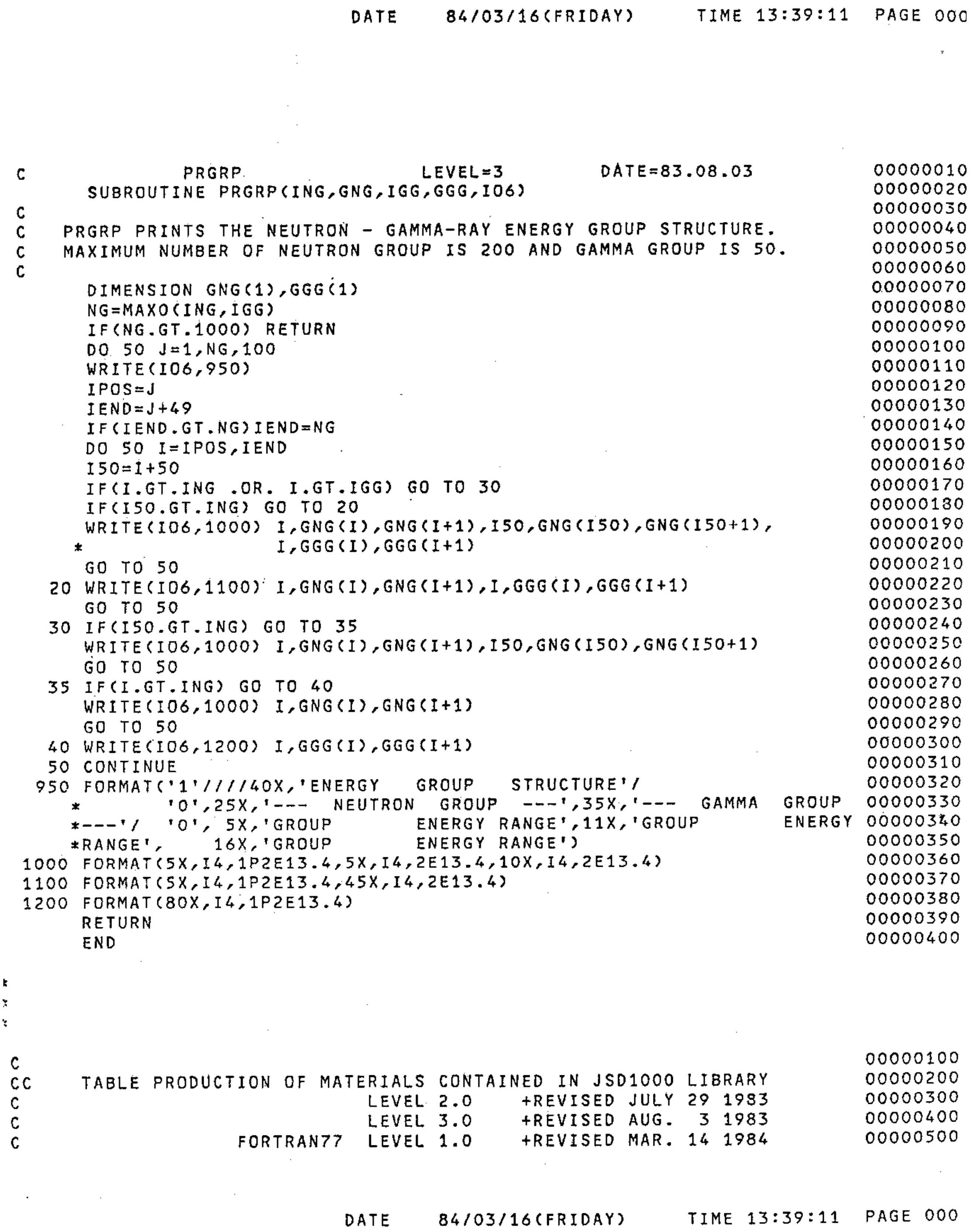
```
                              DATE    84/03/16(FRIDAY)     TIME 13:39:11  PAGE 000:

E  'I-  ','6    ',' 5- ','B- 1','0    ','79-A','U-19','7    ','90-T', 00005100
F  'H-23','2    ','49-I','N-11','5    ','16- ','S- 3','2    ','26-F', 00005200
G  'E- 5','6    ','29-C','U- 6','3    ','29-C','U- 6','5    ','53- ', 00005300
H  'I-12','7    ','21-S','C- 4','5    ','49-I','N-11','5    ','26-F', 00005400
I  'E- 5','4    ','26-F','E- 5','8    ','26-N','I- 5','8    ','28-N', 00005500
J  'I- 6','0    ','22-T','I- 4','6    ','22-T','I- 4','7    '/        00005600
 DATA ((MCR(I,J),I=1,3),J=117,141)/    '22-T','I- 4','8    ','26-F', 00005700
1  'E   ','    ','72-H','F-17','4    ','72-H','F-17','6    ','72-H', 00005800
2  'F-17','7    ','72-H','F-17','8    ','72-H','F-17','9    ','72-H', 00005900
3  'F-18','0    ','   A','R    ','    ','    ','    ','    ','    ', 00006000
4  '    ','    ','    ','    ','    ','    ','    ','    ','    ', 00006100
5  '    ','    ','    ','    ','    ','    ','    ','    ','    ', 00006200
6  '    ','    ','    ','    ','    ','    ','    ','    ','    ', 00006300
7  '    ','    ','    ','    ','    ','    ','    ','    ','    ', 00006400
8  '    ','    ','    ','    ','    ','? ? ','? ? ','?    '/        00006500
 END                                                                 00006600



      SUBROUTINE NUMBRP(X,Y,IH,A,M,L,IC)                             00000100
      CHARACTER KL*1                                                 00000200
      CHARACTER DL*2                                                 00000300
      CHARACTER ZL*4                                                 00000400
      IDEC=0                                                         00000500
      ICHK=0                                                         00000600
      B=A                                                            00000700
      IF(B.EQ.0.0) GO TO 100                                         00000800
      IF(B.GT.0.0) GO TO 20                                          00000900
      KL='-'                                                         00001000
      CALL GSCHAR(X,Y,IH,KL,IC,1)                                    00001100
      B=ABS(B)                                                       00001200
      ICHK=1                                                         00001300
   20 ADEC=ALOG10(B)                                                 00001400
      IF(ADEC.LT.0.) ADEC=ADEC-1                                     00001500
      IDEC=ADEC                                                      00001600
      JDEC=IABS(IDEC)                                                00001700
      IF(IDEC.EQ.0) GO TO 10                                         00001800
      IF(IDEC.GT.0) GO TO 2                                          00001900
      AA=B*10.**JDEC                                                 00002000
      GO TO 3                                                        00002100
    2 AA=B*10./(10.**JDEC)                                           00002200
      GO TO 3                                                        00002300
   10 AA=B                                                           00002400
    3 CONTINUE                                                       00002500
CC         SHI-SHA GO-NYU                                            00002600
      ABT=1.0                                                        00002700
      IF(M.GT.0) ABT=10.**M                                          00002800
      ABL=AA*ABT                                                     00002900
      IABL=IFIX(ABL)                                                 00003000
      ABL=ABL-FLOAT(IABL)                                            00003100
      IF(ABL.GE.0.5) IABL=IABL+1                                     00003200
      JBL=IABL/IFIX(ABT)                                             00003300
      WRITE(KL,1) JBL                                                00003400


                              DATE    84/03/16(FRIDAY)     TIME 13:39:11  PAGE 000
```

Fig. 4.1 (continued)



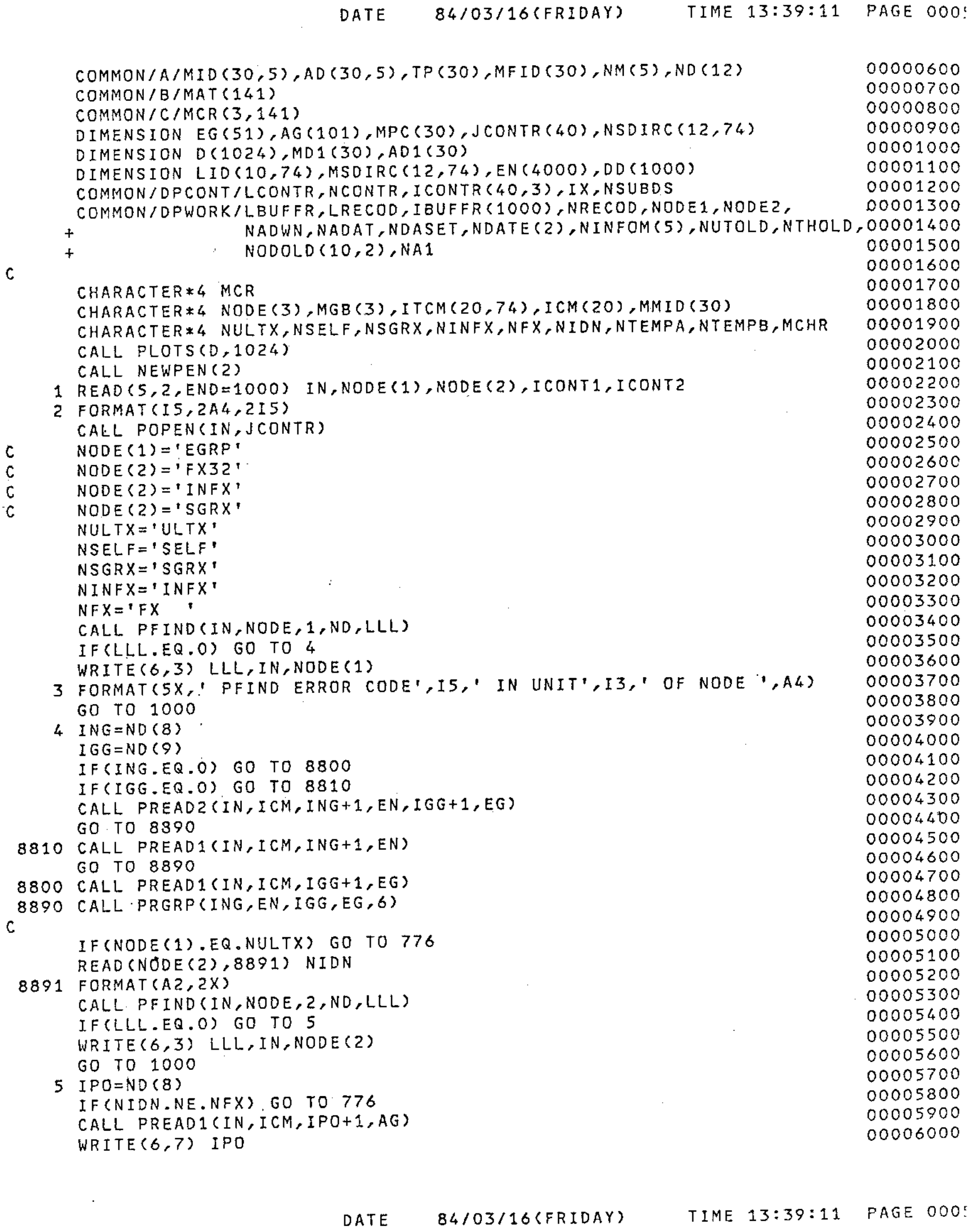
```
                         DATE     84/03/16(FRIDAY)      TIME 13:39:11   PAGE 000

    1 FORMAT(I1)
      XX=999.
      YY=999.
      IF(ICHK.EQ.0) XX=X
      IF(ICHK.EQ.0) YY=Y
      CALL GSCHAR(XX,YY,IH,KL,IC,1)
      IF(M.EQ.0) GO TO 5
      KL='.'
      CALL GSCHAR(999.,999.,IH,KL,IC,1)
      DO 4 I=1,M
      IABL=10*(IABL-JBL*IFIX(ABT))
      JBL=IABL/IFIX(ABT)
      WRITE(KL,1) JBL
      CALL GSCHAR(999.,999.,IH,KL,IC,1)
    4 CONTINUE
    5 CONTINUE
      IF(IDEC.EQ.0) GO TO 1000
      IF(L.EQ.0) GO TO 6
      KL='*'
      CALL GSCHR1(999.,999.,IH,KL,IC,1,9)
      DL='10'
      CALL GSCHAR(999.,999.,IH,DL,IC,2)
      KL='-'
      JH=IH-1
      YYY=Y+2.7
      IF(IDEC.LT.0) CALL GSCHAR(999.,YYY,JH,KL,IC,1)
      IF(JDEC.LT.10) GO TO 8
      WRITE(DL,7)  JDEC
    7 FORMAT(I2)
      CALL GSCHAR(999.,YYY,JH,DL,IC,2)
      GO TO 1000
    8 WRITE(KL,1)  JDEC
      CALL GSCHAR(999.,YYY,JH,KL,IC,1)
      GO TO 1000
    6 CONTINUE
      KL='E'
      CALL GSCHAR(999.,999.,IH,KL,IC,1)
      KL='-'
      IF(IDEC.LT.0) CALL GSCHAR(999.,999.,IH,KL,IC,1)
      IF(JDEC.LT.10) GO TO 9
      WRITE(DL,7) JDEC
      CALL GSCHAR(999.,999.,IH,DL,IC,2)
      GO TO 1000
    9 WRITE(KL,1) JDEC
      CALL GSCHAR(999.,999.,IH,KL,IC,1)
      GO TO 1000
  100 ZL='0.0 '
      CALL GSCHAR(X,Y,IH,ZL,IC,4)
 1000 CONTINUE
      RETURN
      E   N   D

                         DATE     84/03/16(FRIDAY)      TIME 13:39:11   PAGE 000
```

Fig. 4.1 (continued)



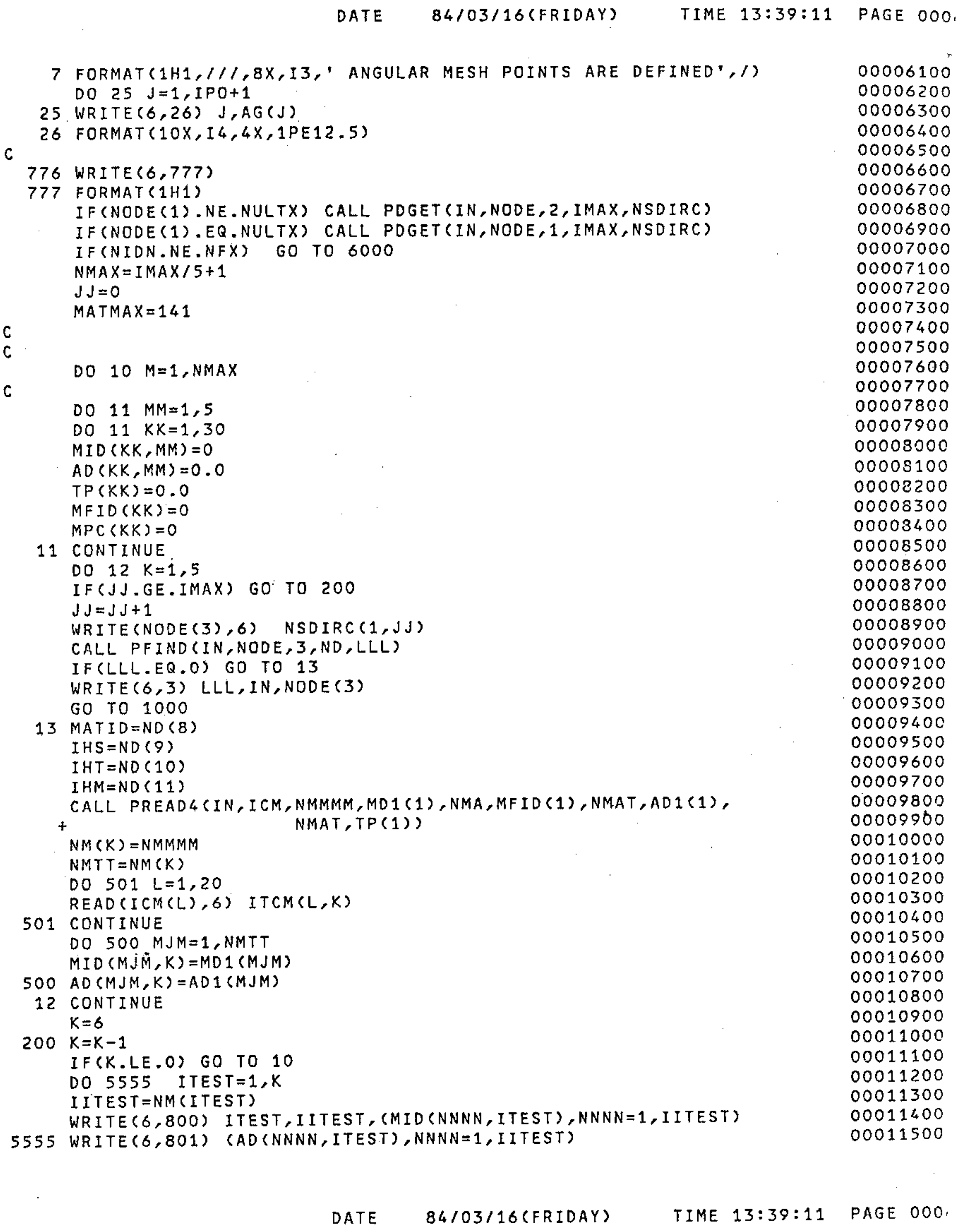
```
                        DATE    84/03/16(FRIDAY)     TIME 13:39:11  PAGE 000




C             PRGRP                   LEVEL=3         DATE=83.08.03
      SUBROUTINE PRGRP(ING,GNG,IGG,GGG,I06)
C
C   PRGRP PRINTS THE NEUTRON - GAMMA-RAY ENERGY GROUP STRUCTURE.
C   MAXIMUM NUMBER OF NEUTRON GROUP IS 200 AND GAMMA GROUP IS 50.
C
      DIMENSION GNG(1),GGG(1)
      NG=MAXO(ING,IGG)
      IF(NG.GT.1000) RETURN
      DO 50 J=1,NG,100
      WRITE(I06,950)
      IPOS=J
      IEND=J+49
      IF(IEND.GT.NG)IEND=NG
      DO 50 I=IPOS,IEND
      I50=I+50
      IF(I.GT.ING .OR. I.GT.IGG) GO TO 30
      IF(I50.GT.ING) GO TO 20
      WRITE(I06,1000) I,GNG(I),GNG(I+1),I50,GNG(I50),GNG(I50+1),
     *                I,GGG(I),GGG(I+1)
      GO TO 50
   20 WRITE(I06,1100) I,GNG(I),GNG(I+1),I,GGG(I),GGG(I+1)
      GO TO 50
   30 IF(I50.GT.ING) GO TO 35
      WRITE(I06,1000) I,GNG(I),GNG(I+1),I50,GNG(I50),GNG(I50+1)
      GO TO 50
   35 IF(I.GT.ING) GO TO 40
      WRITE(I06,1000) I,GNG(I),GNG(I+1)
      GO TO 50
   40 WRITE(I06,1200) I,GGG(I),GGG(I+1)
   50 CONTINUE
  950 FORMAT('1'////40X,'ENERGY   GROUP   STRUCTURE'/
     *       'O',25X,'---  NEUTRON  GROUP  ---',35X,'---  GAMMA  GROUP
     *---'/  'O', 5X,'GROUP       ENERGY RANGE',11X,'GROUP       ENERGY
     *RANGE',    16X,'GROUP       ENERGY RANGE')
 1000 FORMAT(5X,I4,1P2E13.4,5X,I4,2E13.4,10X,I4,2E13.4)
 1100 FORMAT(5X,I4,1P2E13.4,45X,I4,2E13.4)
 1200 FORMAT(80X,I4,1P2E13.4)
      RETURN
      END



C
CC    TABLE PRODUCTION OF MATERIALS CONTAINED IN JSD1000 LIBRARY
C                           LEVEL 2.0    +REVISED JULY 29 1983
C                           LEVEL 3.0    +REVISED AUG.  3 1983
C                FORTRAN77  LEVEL 1.0    +REVISED MAR. 14 1984

                        DATE    84/03/16(FRIDAY)     TIME 13:39:11  PAGE 000
```

Fig. 4.1 (continued)



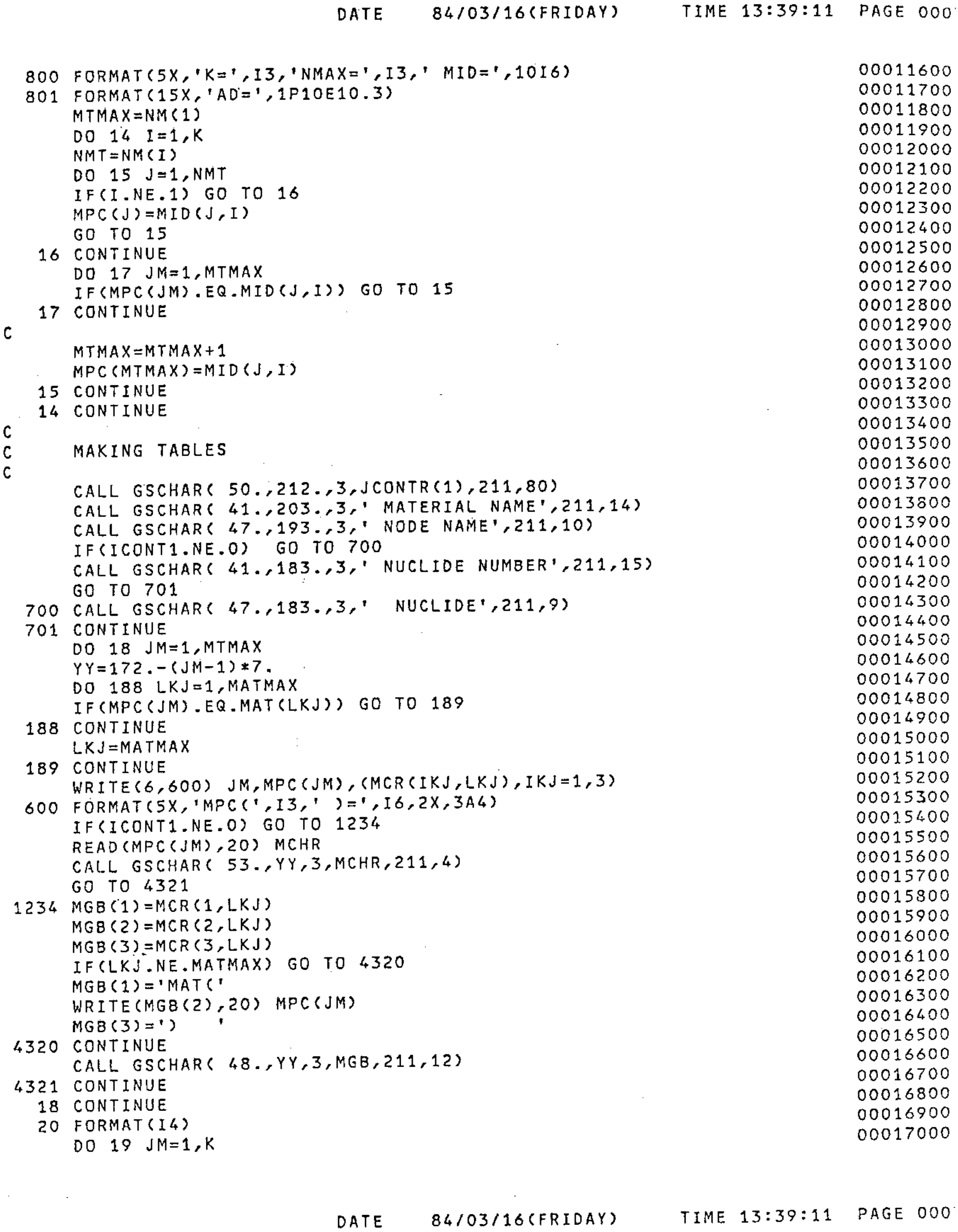
```
                         DATE     84/03/16(FRIDAY)      TIME 13:39:11  PAGE 000!

      COMMON/A/MID(30,5),AD(30,5),TP(30),MFID(30),NM(5),ND(12)
      COMMON/B/MAT(141)
      COMMON/C/MCR(3,141)
      DIMENSION EG(51),AG(101),MPC(30),JCONTR(40),NSDIRC(12,74)
      DIMENSION D(1024),MD1(30),AD1(30)
      DIMENSION LID(10,74),MSDIRC(12,74),EN(4000),DD(1000)
      COMMON/DPCONT/LCONTR,NCONTR,ICONTR(40,3),IX,NSUBDS
      COMMON/DPWORK/LBUFFR,LRECOD,IBUFFR(1000),NRECOD,NODE1,NODE2,
     +              NADWN,NADAT,NDASET,NDATE(2),NINFOM(5),NUTOLD,NTHOLD,
     +              NODOLD(10,2),NA1
C
      CHARACTER*4 MCR
      CHARACTER*4 NODE(3),MGB(3),ITCM(20,74),ICM(20),MMID(30)
      CHARACTER*4 NULTX,NSELF,NSGRX,NINFX,NFX,NIDN,NTEMPA,NTEMPB,MCHR
      CALL PLOTS(D,1024)
      CALL NEWPEN(2)
    1 READ(5,2,END=1000) IN,NODE(1),NODE(2),ICONT1,ICONT2
    2 FORMAT(I5,2A4,2I5)
      CALL POPEN(IN,JCONTR)
C     NODE(1)='EGRP'
C     NODE(2)='FX32'
C     NODE(2)='INFX'
C     NODE(2)='SGRX'
      NULTX='ULTX'
      NSELF='SELF'
      NSGRX='SGRX'
      NINFX='INFX'
      NFX='FX  '
      CALL PFIND(IN,NODE,1,ND,LLL)
      IF(LLL.EQ.0) GO TO 4
      WRITE(6,3) LLL,IN,NODE(1)
    3 FORMAT(5X,' PFIND ERROR CODE',I5,' IN UNIT',I3,' OF NODE ',A4)
      GO TO 1000
    4 ING=ND(8)
      IGG=ND(9)
      IF(ING.EQ.0) GO TO 8800
      IF(IGG.EQ.0) GO TO 8810
      CALL PREAD2(IN,ICM,ING+1,EN,IGG+1,EG)
      GO TO 8890
 8810 CALL PREAD1(IN,ICM,ING+1,EN)
      GO TO 8890
 8800 CALL PREAD1(IN,ICM,IGG+1,EG)
 8890 CALL PRGRP(ING,EN,IGG,EG,6)
C
      IF(NODE(1).EQ.NULTX) GO TO 776
      READ(NODE(2),8891) NIDN
 8891 FORMAT(A2,2X)
      CALL PFIND(IN,NODE,2,ND,LLL)
      IF(LLL.EQ.0) GO TO 5
      WRITE(6,3) LLL,IN,NODE(2)
      GO TO 1000
    5 IPO=ND(8)
      IF(NIDN.NE.NFX) GO TO 776
      CALL PREAD1(IN,ICM,IPO+1,AG)
      WRITE(6,7) IPO

                         DATE     84/03/16(FRIDAY)      TIME 13:39:11  PAGE 000!
```

Fig. 4.1 (continued)



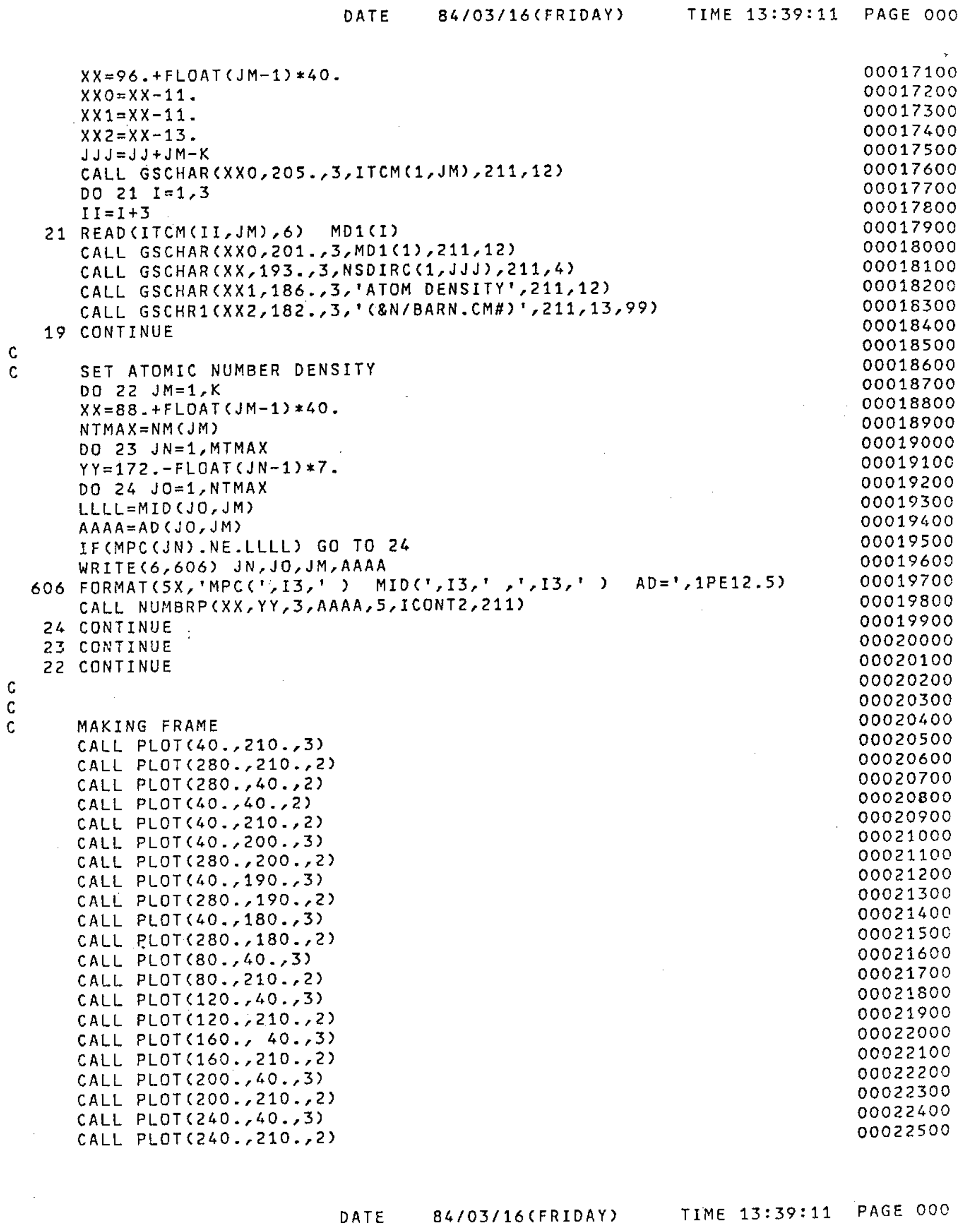
```
                          DATE    84/03/16(FRIDAY)     TIME 13:39:11  PAGE 000

    7 FORMAT(1H1,///,8X,I3,' ANGULAR MESH POINTS ARE DEFINED',/)
      DO 25 J=1,IPO+1
   25 WRITE(6,26) J,AG(J)
   26 FORMAT(10X,I4,4X,1PE12.5)
C
  776 WRITE(6,777)
  777 FORMAT(1H1)
      IF(NODE(1).NE.NULTX) CALL PDGET(IN,NODE,2,IMAX,NSDIRC)
      IF(NODE(1).EQ.NULTX) CALL PDGET(IN,NODE,1,IMAX,NSDIRC)
      IF(NIDN.NE.NFX)  GO TO 6000
      NMAX=IMAX/5+1
      JJ=0
      MATMAX=141
C
C
      DO 10 M=1,NMAX
C
      DO 11 MM=1,5
      DO 11 KK=1,30
      MID(KK,MM)=0
      AD(KK,MM)=0.0
      TP(KK)=0.0
      MFID(KK)=0
      MPC(KK)=0
   11 CONTINUE
      DO 12 K=1,5
      IF(JJ.GE.IMAX) GO TO 200
      JJ=JJ+1
      WRITE(NODE(3),6)  NSDIRC(1,JJ)
      CALL PFIND(IN,NODE,3,ND,LLL)
      IF(LLL.EQ.0) GO TO 13
      WRITE(6,3) LLL,IN,NODE(3)
      GO TO 1000
   13 MATID=ND(8)
      IHS=ND(9)
      IHT=ND(10)
      IHM=ND(11)
      CALL PREAD4(IN,ICM,NMMMM,MD1(1),NMA,MFID(1),NMAT,AD1(1),
     +               NMAT,TP(1))
      NM(K)=NMMMM
      NMTT=NM(K)
      DO 501 L=1,20
      READ(ICM(L),6) ITCM(L,K)
  501 CONTINUE
      DO 500 MJM=1,NMTT
      MID(MJM,K)=MD1(MJM)
  500 AD(MJM,K)=AD1(MJM)
   12 CONTINUE
      K=6
  200 K=K-1
      IF(K.LE.0) GO TO 10
      DO 5555  ITEST=1,K
      IITEST=NM(ITEST)
      WRITE(6,800) ITEST,IITEST,(MID(NNNN,ITEST),NNNN=1,IITEST)
 5555 WRITE(6,801) (AD(NNNN,ITEST),NNNN=1,IITEST)


                          DATE    84/03/16(FRIDAY)     TIME 13:39:11  PAGE 000
```

Fig. 4.1 (continued)



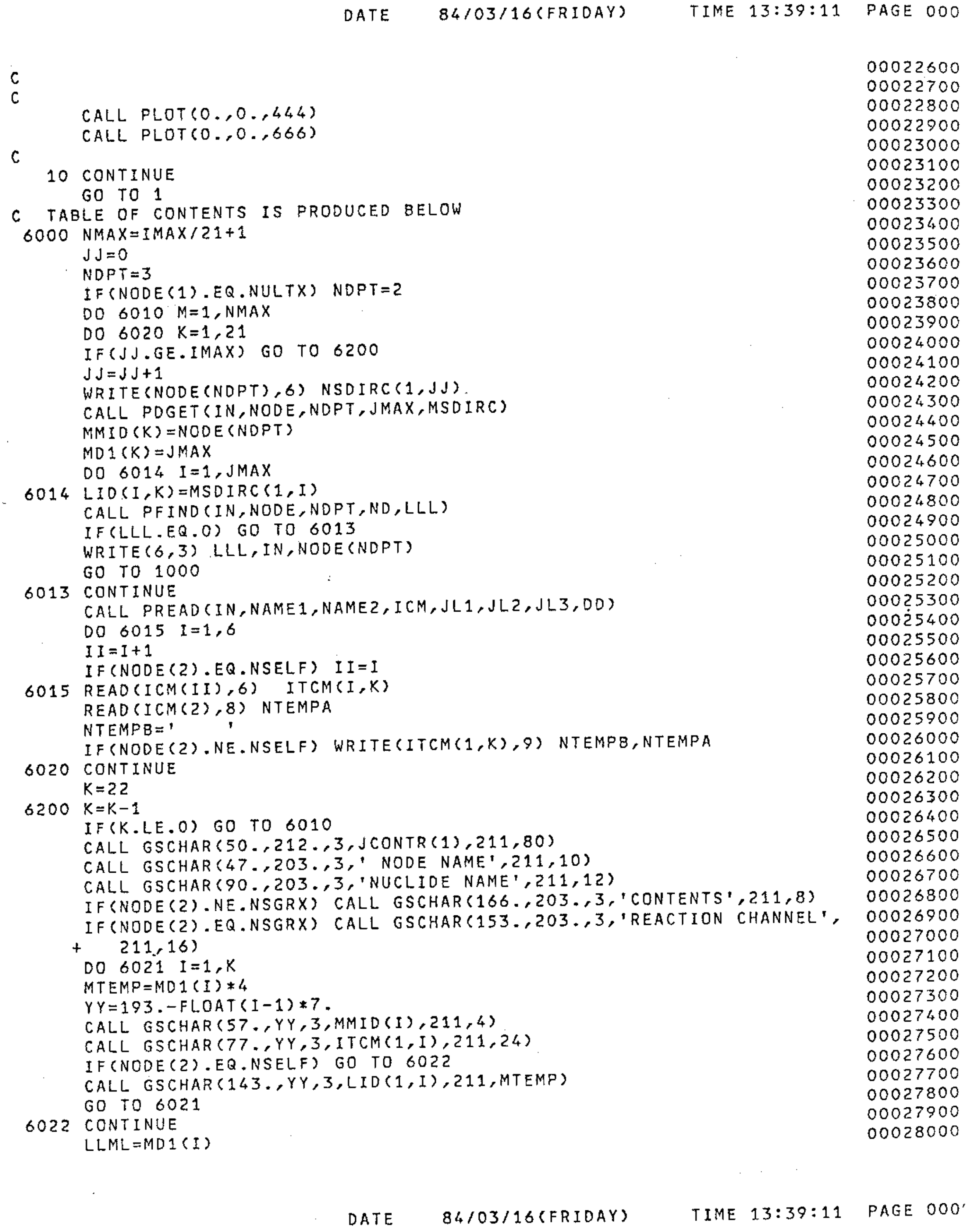
```
                     DATE    84/03/16(FRIDAY)     TIME 13:39:11  PAGE 000

  800 FORMAT(5X,'K=',I3,'NMAX=',I3,' MID=',10I6)
  801 FORMAT(15X,'AD=',1P10E10.3)
      MTMAX=NM(1)
      DO 14 I=1,K
      NMT=NM(I)
      DO 15 J=1,NMT
      IF(I.NE.1) GO TO 16
      MPC(J)=MID(J,I)
      GO TO 15
   16 CONTINUE
      DO 17 JM=1,MTMAX
      IF(MPC(JM).EQ.MID(J,I)) GO TO 15
   17 CONTINUE
C
      MTMAX=MTMAX+1
      MPC(MTMAX)=MID(J,I)
   15 CONTINUE
   14 CONTINUE
C
C     MAKING TABLES
C
      CALL GSCHAR( 50.,212.,3,JCONTR(1),211,80)
      CALL GSCHAR( 41.,203.,3,' MATERIAL NAME',211,14)
      CALL GSCHAR( 47.,193.,3,' NODE NAME',211,10)
      IF(ICONT1.NE.0)  GO TO 700
      CALL GSCHAR( 41.,183.,3,' NUCLIDE NUMBER',211,15)
      GO TO 701
  700 CALL GSCHAR( 47.,183.,3,'  NUCLIDE',211,9)
  701 CONTINUE
      DO 18 JM=1,MTMAX
      YY=172.-(JM-1)*7.
      DO 188 LKJ=1,MATMAX
      IF(MPC(JM).EQ.MAT(LKJ)) GO TO 189
  188 CONTINUE
      LKJ=MATMAX
  189 CONTINUE
      WRITE(6,600) JM,MPC(JM),(MCR(IKJ,LKJ),IKJ=1,3)
  600 FORMAT(5X,'MPC(',I3,' )=',I6,2X,3A4)
      IF(ICONT1.NE.0) GO TO 1234
      READ(MPC(JM),20) MCHR
      CALL GSCHAR( 53.,YY,3,MCHR,211,4)
      GO TO 4321
 1234 MGB(1)=MCR(1,LKJ)
      MGB(2)=MCR(2,LKJ)
      MGB(3)=MCR(3,LKJ)
      IF(LKJ.NE.MATMAX) GO TO 4320
      MGB(1)='MAT('
      WRITE(MGB(2),20) MPC(JM)
      MGB(3)=')    '
 4320 CONTINUE
      CALL GSCHAR( 48.,YY,3,MGB,211,12)
 4321 CONTINUE
   18 CONTINUE
   20 FORMAT(I4)
      DO 19 JM=1,K

                     DATE    84/03/16(FRIDAY)     TIME 13:39:11  PAGE 000
```

Fig. 4.1 (continued)



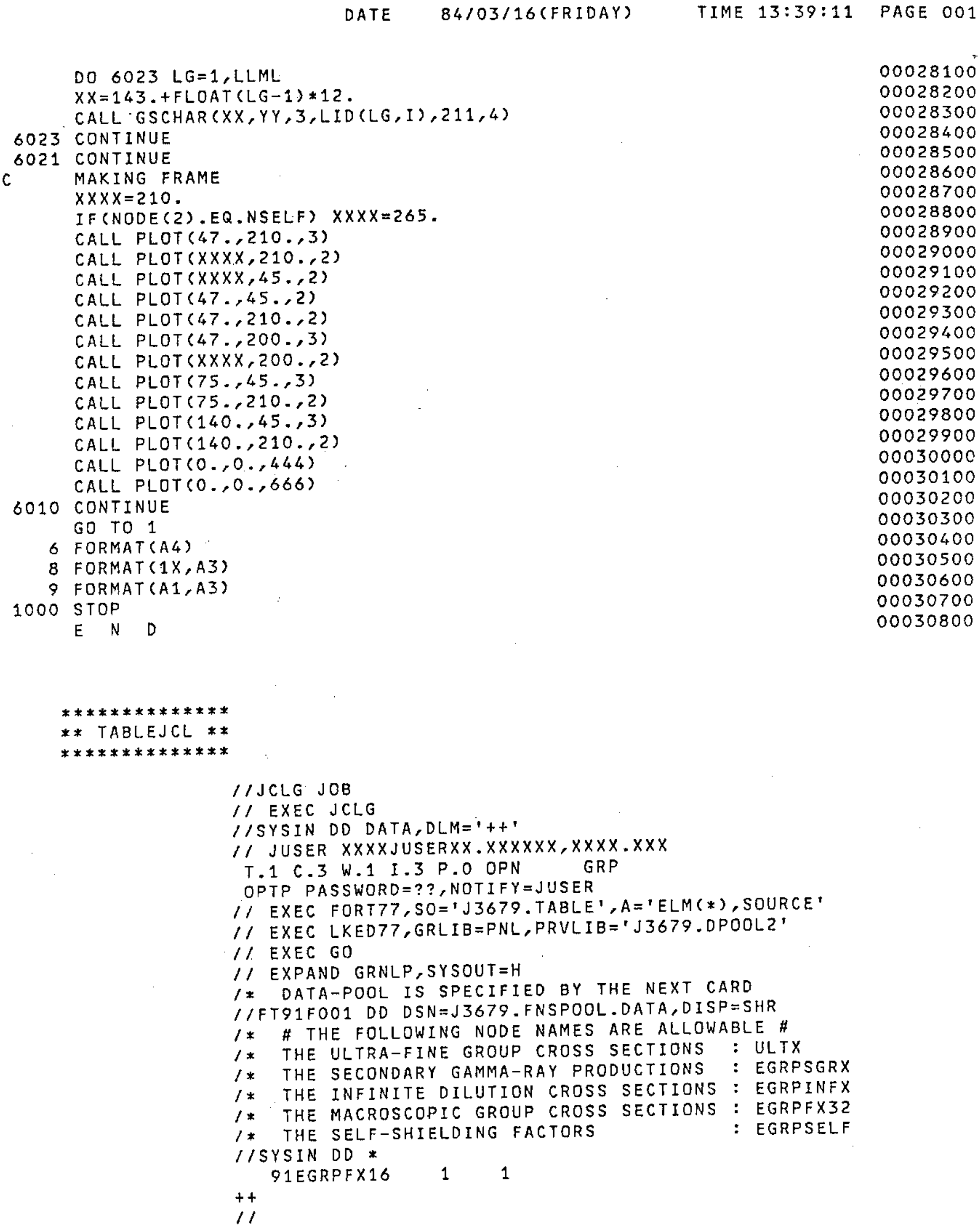
```
                        DATE     84/03/16(FRIDAY)      TIME 13:39:11  PAGE 000

      XX=96.+FLOAT(JM-1)*40.
      XX0=XX-11.
      XX1=XX-11.
      XX2=XX-13.
      JJJ=JJ+JM-K
      CALL GSCHAR(XX0,205.,3,ITCM(1,JM),211,12)
      DO 21 I=1,3
      II=I+3
   21 READ(ITCM(II,JM),6)  MD1(I)
      CALL GSCHAR(XX0,201.,3,MD1(1),211,12)
      CALL GSCHAR(XX,193.,3,NSDIRC(1,JJJ),211,4)
      CALL GSCHAR(XX1,186.,3,'ATOM DENSITY',211,12)
      CALL GSCHR1(XX2,182.,3,'(&N/BARN.CM#)',211,13,99)
   19 CONTINUE
C
C     SET ATOMIC NUMBER DENSITY
      DO 22 JM=1,K
      XX=88.+FLOAT(JM-1)*40.
      NTMAX=NM(JM)
      DO 23 JN=1,MTMAX
      YY=172.-FLOAT(JN-1)*7.
      DO 24 JO=1,NTMAX
      LLLL=MID(JO,JM)
      AAAA=AD(JO,JM)
      IF(MPC(JN).NE.LLLL) GO TO 24
      WRITE(6,606) JN,JO,JM,AAAA
  606 FORMAT(5X,'MPC(',I3,' )  MID(',I3,' ,',I3,' )  AD=',1PE12.5)
      CALL NUMBRP(XX,YY,3,AAAA,5,ICONT2,211)
   24 CONTINUE
   23 CONTINUE
   22 CONTINUE
C
C
C     MAKING FRAME
      CALL PLOT(40.,210.,3)
      CALL PLOT(280.,210.,2)
      CALL PLOT(280.,40.,2)
      CALL PLOT(40.,40.,2)
      CALL PLOT(40.,210.,2)
      CALL PLOT(40.,200.,3)
      CALL PLOT(280.,200.,2)
      CALL PLOT(40.,190.,3)
      CALL PLOT(280.,190.,2)
      CALL PLOT(40.,180.,3)
      CALL PLOT(280.,180.,2)
      CALL PLOT(80.,40.,3)
      CALL PLOT(80.,210.,2)
      CALL PLOT(120.,40.,3)
      CALL PLOT(120.,210.,2)
      CALL PLOT(160., 40.,3)
      CALL PLOT(160.,210.,2)
      CALL PLOT(200.,40.,3)
      CALL PLOT(200.,210.,2)
      CALL PLOT(240.,40.,3)
      CALL PLOT(240.,210.,2)

                        DATE     84/03/16(FRIDAY)      TIME 13:39:11  PAGE 000
```

Fig. 4.1 (continued)



```
                    DATE    84/03/16(FRIDAY)     TIME 13:39:11  PAGE 000

C
C
      CALL PLOT(0.,0.,444)
      CALL PLOT(0.,0.,666)
C
   10 CONTINUE
      GO TO 1
C  TABLE OF CONTENTS IS PRODUCED BELOW
 6000 NMAX=IMAX/21+1
      JJ=0
      NDPT=3
      IF(NODE(1).EQ.NULTX) NDPT=2
      DO 6010 M=1,NMAX
      DO 6020 K=1,21
      IF(JJ.GE.IMAX) GO TO 6200
      JJ=JJ+1
      WRITE(NODE(NDPT),6) NSDIRC(1,JJ)
      CALL PDGET(IN,NODE,NDPT,JMAX,MSDIRC)
      MMID(K)=NODE(NDPT)
      MD1(K)=JMAX
      DO 6014 I=1,JMAX
 6014 LID(I,K)=MSDIRC(1,I)
      CALL PFIND(IN,NODE,NDPT,ND,LLL)
      IF(LLL.EQ.0) GO TO 6013
      WRITE(6,3) LLL,IN,NODE(NDPT)
      GO TO 1000
 6013 CONTINUE
      CALL PREAD(IN,NAME1,NAME2,ICM,JL1,JL2,JL3,DD)
      DO 6015 I=1,6
      II=I+1
      IF(NODE(2).EQ.NSELF) II=I
 6015 READ(ICM(II),6)  ITCM(I,K)
      READ(ICM(2),8) NTEMPA
      NTEMPB='    '
      IF(NODE(2).NE.NSELF) WRITE(ITCM(1,K),9) NTEMPB,NTEMPA
 6020 CONTINUE
      K=22
 6200 K=K-1
      IF(K.LE.0) GO TO 6010
      CALL GSCHAR(50.,212.,3,JCONTR(1),211,80)
      CALL GSCHAR(47.,203.,3,' NODE NAME',211,10)
      CALL GSCHAR(90.,203.,3,'NUCLIDE NAME',211,12)
      IF(NODE(2).NE.NSGRX) CALL GSCHAR(166.,203.,3,'CONTENTS',211,8)
      IF(NODE(2).EQ.NSGRX) CALL GSCHAR(153.,203.,3,'REACTION CHANNEL',
     +   211,16)
      DO 6021 I=1,K
      MTEMP=MD1(I)*4
      YY=193.-FLOAT(I-1)*7.
      CALL GSCHAR(57.,YY,3,MMID(I),211,4)
      CALL GSCHAR(77.,YY,3,ITCM(1,I),211,24)
      IF(NODE(2).EQ.NSELF) GO TO 6022
      CALL GSCHAR(143.,YY,3,LID(1,I),211,MTEMP)
      GO TO 6021
 6022 CONTINUE
      LLML=MD1(I)

                    DATE    84/03/16(FRIDAY)     TIME 13:39:11  PAGE 000'
```

Fig. 4.1 (continued)



```
                              DATE    84/03/16(FRIDAY)     TIME 13:39:11  PAGE 001

       DO 6023 LG=1,LLML
       XX=143.+FLOAT(LG-1)*12.
       CALL GSCHAR(XX,YY,3,LID(LG,I),211,4)
 6023  CONTINUE
 6021  CONTINUE
C      MAKING FRAME
       XXXX=210.
       IF(NODE(2).EQ.NSELF) XXXX=265.
       CALL PLOT(47.,210.,3)
       CALL PLOT(XXXX,210.,2)
       CALL PLOT(XXXX,45.,2)
       CALL PLOT(47.,45.,2)
       CALL PLOT(47.,210.,2)
       CALL PLOT(47.,200.,3)
       CALL PLOT(XXXX,200.,2)
       CALL PLOT(75.,45.,3)
       CALL PLOT(75.,210.,2)
       CALL PLOT(140.,45.,3)
       CALL PLOT(140.,210.,2)
       CALL PLOT(0.,0.,444)
       CALL PLOT(0.,0.,666)
 6010  CONTINUE
       GO TO 1
    6  FORMAT(A4)
    8  FORMAT(1X,A3)
    9  FORMAT(A1,A3)
 1000  STOP
       E  N  D


      **************
      ** TABLEJCL **
      **************

                   //JCLG JOB
                   // EXEC JCLG
                   //SYSIN DD DATA,DLM='++'
                   // JUSER XXXXJUSERXX.XXXXXX,XXXX.XXX
                    T.1 C.3 W.1 I.3 P.0 OPN     GRP
                    OPTP PASSWORD=??,NOTIFY=JUSER
                   // EXEC FORT77,SO='J3679.TABLE',A='ELM(*),SOURCE'
                   // EXEC LKED77,GRLIB=PNL,PRVLIB='J3679.DPOOL2'
                   // EXEC GO
                   // EXPAND GRNLP,SYSOUT=H
                   /*  DATA-POOL IS SPECIFIED BY THE NEXT CARD
                   //FT91F001 DD DSN=J3679.FNSPOOL.DATA,DISP=SHR
                   /*  # THE FOLLOWING NODE NAMES ARE ALLOWABLE #
                   /*  THE ULTRA-FINE GROUP CROSS SECTIONS   : ULTX
                   /*  THE SECONDARY GAMMA-RAY PRODUCTIONS   : EGRPSGRX
                   /*  THE INFINITE DILUTION CROSS SECTIONS  : EGRPINFX
                   /*  THE MACROSCOPIC GROUP CROSS SECTIONS  : EGRPFX32
                   /*  THE SELF-SHIELDING FACTORS            : EGRPSELF
                   //SYSIN DD *
                     91EGRPFX16     1    1
                   ++
                   //
```

Fig. 4.1 (continued)

5. TSS Management Utility of DATA-POOL

5.1 Outline of POOL

The DATA-POOL is a special direct-access data base, so that the management utility POOL is prepared to maintain the data set. POOL has 14 operations commands and can be executed by TSS terminals or batch jobs.

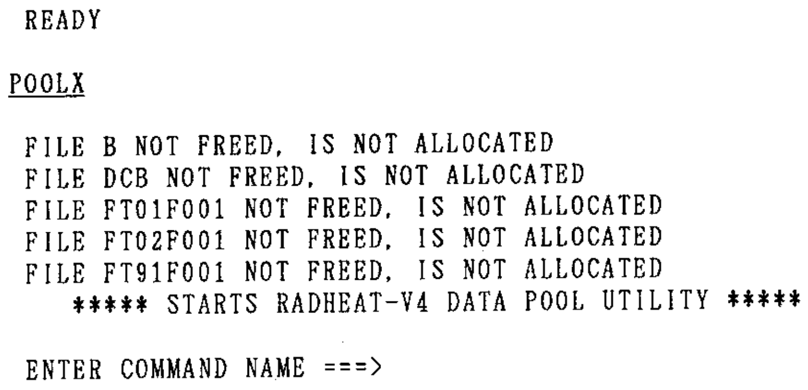
On the TSS terminal, the next operation is necessary to execute the utility POOL. The statement with underline shows the user's input on TSS terminal. CR means a carriage return.

```
READY
EX  'J1446.TSSMAC.CLIST (POOLX)' CR
```

The user who has a data set named TSSMAC.CLIST(POOLX) can execute POOL by a simple operation. The "TSSMAC.CLIST" is a default data-set name consisted of user's cataloged procedures in the JAERI Computing Center.

```
READY
POOLX  CR
```

The cataloged procedure is excuted, the next display appears.



```
READY

POOLX

FILE B NOT FREED, IS NOT ALLOCATED
FILE DCB NOT FREED, IS NOT ALLOCATED
FILE FT01F001 NOT FREED, IS NOT ALLOCATED
FILE FT02F001 NOT FREED, IS NOT ALLOCATED
FILE FT91F001 NOT FREED, IS NOT ALLOCATED
   ***** STARTS RADHEAT-V4 DATA POOL UTILITY *****

ENTER COMMAND NAME ===>
```

The 12 functions of POOL can be selected by the user to enter a command name. HELP command is prepared when the user forgets command names. END command is used for terminating the execution. HELP command is entered, the next display appears.

```
ENTER COMMAND NAME ===> HELP

COMMAND                CONTENTS
CATL      PRINT OF CONTROL AND DIRECTORY SECTION
CONDENSE  CONDENSE OF A DATA POOL
COPY      COPY OF A NODE DATA
DELETE    DELETE OF A NODE DATA
FLAG      CHANGE OF A WRITE FLAG
INIT      INITIALIZATION OF A DATA POOL
LIST      LISTING OF A NODE DATA (SUB-DIRECTORY AND FORM OF DATA ARRAYS)
MEND      MENDING OF A CONTROL, DIRECTORY AND DATA COMMENT
MTCOPY    LOAD OF A BACK-UP TAPE TO A DATA POOL
MTSAVE    MAKING OF A BACK-UP TAPE
RENAME    RENAME OF A NODE
TREE      PRINT OF ALL NODE NAMES IN A DATA POOL BY A TREE STRUCTURE

ENTER COMMAND NAME ===>
```

Some commands have the abbreviated forms as follows :

FLAG → F
DELETE → DEL
TREE → T
CATL → C
LIST → L
RENAME → RE
CONDENSE → COND

The functions and usages of these commands are described in the following Sections. The operating method for batch job is described in Section 5.14. The cataloged procedure is shown in Section 5.15. The program information of the POOL is noted in Section 5.16.

5.2 INIT Command

DATA-POOL is initialized by using the INIT command. The initialization is executed by the conversational data inputs with TSS terminal. The initialization for the existing DATA-POOL means erasing the all of data from the DATA-POOL. The sample procedure is as follows :

```
ENTER COMMAND NAME ===> INIT

ENTER DSN OF DATA POOL ===> J3679.DATAPOOL.DATA
ALLOCATION OF DATA SET (NEW/OLD) ====> NEW
UNIT PARAMETER            ==============> TSSWK
SPACE PARAMETER (1-ST SPACE)     ======> 50
SPACE PARAMETER (INCREMENT)      ======> 10
SPACE PARAMETER (SPACE UNIT T/CY) ===> T
ENTER DIRECTORY SIZE
01100 ?
          20
ENTER TITLE (64 CHARACTERS)
01500 ?
        RADHEAT-V4 DATA-POOL FOR CROSS SECTIONS STORAGE
*** MESSAGE FROM PINIT ***
    NO. OF INITIALIZED RECORD IS 1000
*********   C O N T R O L  S E C T I O N   *********
     COL.
    1-18  TITLE   :
               RADHEAT-V4 DATA-POOL FOR CROSS SECTIONS STORAGE
      21  ADDRESS FOR THE DIRECTORY OF FIRST LEVEL NODE  :    2
      22  HEAD ADDRESS FOR THE VACANT DIRECTORY AREA     :    3
      23  HEAD ADDRESS FOR THE VACANT DATA AREA          :   22
      24  WRITE FLAG                                     :    0
      25  READ FLAG (NOT USED)                           :    0
      26  LENGTH OF THE ONE PHYSICAL RECORD              :  900
      27  MAXIMUM NUMBER OF THE SAME LEVEL NODE          :   74
      28  SIZE OF THE DIRECTORY SECTION                  :   20
      29  SIZE OF THE DATA SECTION                       :  979
      30  REAL NUMBER OF THE DIRECTORY RECORDS           :    1
      31  REAL NUMBER OF THE DATA SET RECORDS            :    0
ERROR SUMMARY (FORTRAN77)
ERROR NUMBER ERROR COUNT
    232       001
```

In the sample, the UNIT PARAMETER means the unit name defined in a DD statement such as D0950B, TDS, TSSWK, MSS. The SPACE UNIT of T(track=19k bytes) or CY(Cylinder=250k bytes) can be specified. After the specifications for the data-set, two input data of a directory size and a comment (64-characters) are required. When the initialization is terminated, the Control Section is displayed and the intialized records can be known.
An error of the error number 232 is prearranged one, so that the user is not necessary to pay attention. The data-set name must be specified as a full name because the abbreviated form may cause an error. Note that the volume number and the group number are also required when the allocation for MSS is specified. The allocation space for MSS may be a multiple of cylinder as follows :

```
ENTER COMMAND NAME ===> INIT

ENTER DSN OF DATA POOL ===> J3679.MSSDPOOL.DATA
ALLOCATION OF DATA SET (NEW/OLD) ====> NEW
UNIT PARAMETER          ==============> MSS
SPACE PARAMETER (1-ST SPACE)    ======> 10
SPACE PARAMETER (INCREMENT)     ======> 2
SPACE PARAMETER (SPACE UNIT T/CY) ===> CY
MSS GROUP          ====================> MSS04
MSS VOLUME NUMBER       ==============> MA0072
```

5.3 FLAG Command

The control flag for the exclusive access of the DATA-POOL in the Control Section is set zero by using the FLAG command as follows :

```
ENTER COMMAND NAME ===> FLAG

ENTER DSN OF DATA POOL ===> J3679.DATAPOOL.DATA
 CURRENT STATUS OF WRITE FLAG = 0
 NOW CHANGE WRITE FLAG TO 0
```

5.4 DELETE Command

The node name and data are deleted by using the DELETE command. A Sample is as follows :

```
 ENTER COMMAND NAME ===> DELETE

 ENTER DSN OF DATA POOL ===> J3679.TEST00.DATA
 ENTER NODE NAME
 00900 ?
HA92.SELF.FEE4
 NORMAL RETURN  ***  NODE NAME = HA92.SELF.FEE4.
 ENTER NODE NAME
 00900 ?
BAD .NODE.NAME
 ABNORMAL RETURN  ***  NODE NAME = BAD .NODE.NAME.
 ENTER NODE NAME
 00900 ?
/*
```

The node name to be delete must be specified by the form such as "NOD1.NOD2.NOD3.NOD4" from the first column. A period between two node names must be specified. All directories and data with the last node name, and the all lower levels of this last node name specified by the user are erased. When the process is successfully ended, "NORMAL RETURN" is displayed. However, the process is not ended completely, "ABNORMAL RETURN" is displayed.

To terminate the process, /* CR or CR should be entered.

5.5 TREE Command

The TREE command displays tree structures of node names as follows :

```
ENTER COMMAND NAME ===> TREE

ENTER DSN OF DATA POOL ===> J3679.TEST00.DATA

                   N O D E   T R E E
TITLE OF A DATA POOL          ***
               RADHEAT-V4 DATA-POOL FOR SKYSHINE CALCULATION
LENGTH OF A RECORD                     ***     900
MAXIMUM NUMBER OF THE SAME LEVEL NODE  ***      74
SIZE OF THE DIRECTORY SECTION          ***      40 (USED RECORDS  10)
SIZE OF THE DATA SECTION               ***     928 (USED RECORDS 391)
REMAINS OF THE DIRECTORY SECTION       ***      29
REMAINS OF THE DATA SECTION            ***     534

LEVEL    1       2       3       4       5       6       7       8

    NAGE :  ENERGY GROUP STRUCTURE
     I
     I------INFX :INFINITE DILUTION CROSS SECTION LIBRARY
             I
             I-----1276 :1276 O FROM ENDF/B-IV (300K)
                    I
                    I----- SMT : SMOOTH CROSS SECTIONS
                    I----- FTB : F-TABLE LIBRARY
                    I----- ELA : ELASTIC SCATTERING MATRIX
                    I----- INS : INELASTIC SCATTERING MATRIX

    G09  :  ENERGY GROUP STRUCTURE
     I
     I------TEST :TEST GAMMA-RAY SKYSHINE
             I
             I-----SFX2 :TEST GAMMA-RAY SKYSHINE
             I-----AFX2 :TEST GAMMA-RAY SKYSHINE

    HA92 :  ENERGY GROUP STRUCTURE
     I
     I------SELF : SELF-SHIELDING FACTOR
     I------FX16 : ANGULAR MESH
             I
             I-----FEE4 :IRON ENDF/B-IV MACRO X-SEC. 92G
     I
     I------1010 :NO.101 IRON (0.9MFP) SPHERE (30DEG)
             I
             I-----SFX0 :NO.101 IRON (0.9MFP) SPHERE (30DEG)
```

5.6 CATL Command

The CATL command displays the information of the Directory Section. The sample shown below is corresponded to the node structures of the previous Section. In the sample, "INDEX" means the address of physical record. The node name shown as "////" means that the directory was erased. The other variables are referred to the Section 2.

```
 ENTER COMMAND NAME ===> CATL

 ENTER DSN OF DATA POOL ===> J3679.TEST00.DATA
                                                  D I R E C T O R Y  L I S T
*********   C O N T R O L   S E C T I O N   *********
     COL.
    1-18  TITLE   :
               RADHEAT-V4 DATA-POOL FOR SKYSHINE CALCULATION
      21  ADDRESS FOR THE DIRECTORY OF FIRST LEVEL NODE  :    2
      22  HEAD ADDRESS FOR THE VACANT DIRECTORY AREA     :   13
      23  HEAD ADDRESS FOR THE VACANT DATA AREA          :  436
      24  WRITE FLAG                                     :    0
      25  READ FLAG (NOT USED)                            :    0
      26  LENGTH OF THE ONE PHYSICAL RECORD              :  900
      27  MAXIMUM NUMBER OF THE SAME LEVEL NODE          :   74
      28  SIZE OF THE DIRECTORY SECTION                  :   40
      29  SIZE OF THE DATA SECTION                       :  928
      30  REAL NUMBER OF THE DIRECTORY RECORDS           :   10
      31  REAL NUMBER OF THE DATA SET RECORDS            :  391
*********   D I R E C T O R Y   S E C T I O N   *********
 *** INDEX =     2 ***
        NODE NAME =
        ADDRESS FOR THE UPPER NODE DIRECTORY =  0
        NUMBER OF THE LOWER NODE             =  3
   NO.    NODE   NRECS      NADWN      NADAT      NDASET      DATE
     1    NAGE       1          3         42           1   84-01-20
            INFOM(1)      INFOM(2)      INFOM(3)      INFOM(4)      INFOM(5)
                102             0             0             0             0
   NO.    NODE   NRECS      NADWN      NADAT      NDASET      DATE
     2    G09        1          6        134           1   84-01-24
            INFOM(1)      INFOM(2)      INFOM(3)      INFOM(4)      INFOM(5)
                  0             9             0             0             0
   NO.    NODE   NRECS      NADWN      NADAT      NDASET      DATE
     3    HA92       1          8        291           1   84-01-26
            INFOM(1)      INFOM(2)      INFOM(3)      INFOM(4)      INFOM(5)
                 92             0             0             0             0
 *** INDEX =     3 ***
        NODE NAME = NAGE
        ADDRESS FOR THE UPPER NODE DIRECTORY =  2
        NUMBER OF THE LOWER NODE             =  1
   NO.    NODE   NRECS      NADWN      NADAT      NDASET      DATE
     1    INFX       1          4         43           1   84-01-20
            INFOM(1)      INFOM(2)      INFOM(3)      INFOM(4)      INFOM(5)
                  0             0             0             0             0
```

```
*** INDEX =     4 ***
        NODE NAME = INFX
        ADDRESS FOR THE UPPER NODE DIRECTORY =  3
        NUMBER OF THE LOWER NODE             =  1
  NO.     NODE   NRECS      NADWN      NADAT     NDASET      DATE
   1     1276      1          5         44          1    84-01-20
              INFOM(1)      INFOM(2)      INFOM(3)      INFOM(4)      INFOM(5)
                 1276             0             0             0             0
*** INDEX =     5 ***
        NODE NAME = 1276
        ADDRESS FOR THE UPPER NODE DIRECTORY =  4
        NUMBER OF THE LOWER NODE             =  4
  NO.     NODE   NRECS      NADWN      NADAT     NDASET      DATE
   1       SMT       3          0         45          2  84-01-20
              INFOM(1)      INFOM(2)      INFOM(3)      INFOM(4)      INFOM(5)
                    0             0             0             0             0
  NO.     NODE   NRECS      NADWN      NADAT     NDASET      DATE
   2       FTB       4          0         48          2  84-01-20
              INFOM(1)      INFOM(2)      INFOM(3)      INFOM(4)      INFOM(5)
                    0             0             0             0             0
  NO.     NODE   NRECS      NADWN      NADAT     NDASET      DATE
   3       ELA      57          0         52         12  84-01-20
              INFOM(1)      INFOM(2)      INFOM(3)      INFOM(4)      INFOM(5)
                   18             0             0             0             0
  NO.     NODE   NRECS      NADWN      NADAT     NDASET      DATE
   4       INS      25          0        109         12  84-01-20
              INFOM(1)      INFOM(2)      INFOM(3)      INFOM(4)      INFOM(5)
                    0             0             0             0             0
*** INDEX =     6 ***
        NODE NAME = G09
        ADDRESS FOR THE UPPER NODE DIRECTORY =  2
        NUMBER OF THE LOWER NODE             =  1
  NO.     NODE   NRECS      NADWN      NADAT     NDASET      DATE
   1     TEST        2          7        135          1  84-01-24
              INFOM(1)      INFOM(2)      INFOM(3)      INFOM(4)      INFOM(5)
                    5            41            29             2            48
*** INDEX =     7 ***
        NODE NAME = TEST
        ADDRESS FOR THE UPPER NODE DIRECTORY =  6
        NUMBER OF THE LOWER NODE             =  2
  NO.     NODE   NRECS      NADWN      NADAT     NDASET      DATE
   1     SFX2       18          0        137          9  84-01-24
              INFOM(1)      INFOM(2)      INFOM(3)      INFOM(4)      INFOM(5)
                    0             9             0             0             0
  NO.     NODE   NRECS      NADWN      NADAT     NDASET      DATE
   2     AFX2      136          0        155         46  84-01-24
              INFOM(1)      INFOM(2)      INFOM(3)      INFOM(4)      INFOM(5)
                    0             9             0            48             5
```

```
*** INDEX =    8 ***
       NODE NAME = HA92
       ADDRESS FOR THE UPPER NODE DIRECTORY =  2
       NUMBER OF THE LOWER NODE             =  3
  NO.    NODE   NRECS      NADWN      NADAT      NDASET      DATE
   1     SELF       1          9        292           1   84-01-26
            INFOM(1)      INFOM(2)      INFOM(3)      INFOM(4)      INFOM(5)
                 0             0             0             0             0
  NO.    NODE   NRECS      NADWN      NADAT      NDASET      DATE
   2     FX16       1         11        296           1   84-01-26
            INFOM(1)      INFOM(2)      INFOM(3)      INFOM(4)      INFOM(5)
                16             0             0             0             0
  NO.    NODE   NRECS      NADWN      NADAT      NDASET      DATE
   3     1010       1         12        434           1   84-01-26
            INFOM(1)      INFOM(2)      INFOM(3)      INFOM(4)      INFOM(5)
                 3             9             1             2            16
*** INDEX =    9 ***
       NODE NAME = SELF
       ADDRESS FOR THE UPPER NODE DIRECTORY =  8
       NUMBER OF THE LOWER NODE             =  0
*** INDEX =   10 ***
       NODE NAME = ////
       ADDRESS FOR THE UPPER NODE DIRECTORY =  9
       NUMBER OF THE LOWER NODE             =  2
  NO.    NODE   NRECS      NADWN      NADAT      NDASET      DATE
   1     1192       1          0        294           1   84-01-26
            INFOM(1)      INFOM(2)      INFOM(3)      INFOM(4)      INFOM(5)
               FEE4            2             4             0             0
  NO.    NODE   NRECS      NADWN      NADAT      NDASET      DATE
   2     1274       1          0        295           1   84-01-26
            INFOM(1)      INFOM(2)      INFOM(3)      INFOM(4)      INFOM(5)
               FEE4            2             4             0             0
*** INDEX =   11 ***
       NODE NAME = FX16
       ADDRESS FOR THE UPPER NODE DIRECTORY =  8
       NUMBER OF THE LOWER NODE             =  1
  NO.    NODE   NRECS      NADWN      NADAT      NDASET      DATE
   1     FEE4     137          0        297          93   84-01-26
            INFOM(1)      INFOM(2)      INFOM(3)      INFOM(4)      INFOM(5)
               FEE4            4             3            95             0
*** INDEX =   12 ***
       NODE NAME = 1010
       ADDRESS FOR THE UPPER NODE DIRECTORY =  8
       NUMBER OF THE LOWER NODE             =  1
  NO.    NODE   NRECS      NADWN      NADAT      NDASET      DATE
   1     SFX0       1          0        435           1   84-01-26
            INFOM(1)      INFOM(2)      INFOM(3)      INFOM(4)      INFOM(5)
                92             0             0             0             0
```

5.7 LIST Command

The LIST command displays the record information for the node name specified by the user. A sample is as follows :

```
 ENTER COMMAND NAME ===> LIST

 ENTER DSN OF DATA POOL ===> J3679.TESTOO.DATA
 ENTER NODE NAME
 00900 ?
NAGE.INFX.1276
 RECORD INFORMATION FOR NODE NAME NAGE.INFX.1276.
  ITEM          CONTENTS
     1  NODE NAME              =    1276
     2  TOTAL LENG. OF DATA SET =       1
     3  ADDRESS OF A LOWER NODE =       5
     4  ADDRESS OF A DATA SET   =      44
     5  NO. OF SUB-DATA SETS    =       1
     6  DATE OF CREATION       =84-01-20
     8  DATA 1 =           1276
     9  DATA 2 =              0
    10  DATA 3 =              0
    11  DATA 4 =              0
    12  DATA 5 =              0
 ** INFORMATION FOR SUB-DATA SET   1 **
    1276 0 FROM ENDF/B-IV (300K)
    NO. OF DATA ARRAYS =  1
    LENGTH OF DATA 1=    10  LENGTH OF DATA
```

5.8 RENAME Command

The node names can be renamed by using the RENAME command. The renaming is performed to the last node name specified by the user. To terminate the process, /*$\boxed{\text{CR}}$ or $\boxed{\text{CR}}$ should be entered.
A sample is as follows :

```
 ENTER COMMAND NAME ===> RENAME

 ENTER DSN OF DATA POOL ===> J3679.TESTOO.DATA
 ENTER A OLD NODE NAME
 00900 ?
HA92.SELF
 ENTER A NEW NODE NAME
 00900 ?
HA92.COOP
 RENAME IS FINISHED SUCCESSFULLY
 ENTER A OLD NODE NAME
 00900 ?
/*
```

5.9 COPY Command

The copy from a DATA-POOL to the other DATA-POOL can be performed by using the COPY command. When "*ALL" is entered as a node name, all of data contained in a DATA-POOL are copied. If the node names are entered, data with the last level and the lower levels of nodes are copied. When the node name is already existed in the second data-set, the copy will not execute. The second data set should be initialized when the data set is newly created. To terminate the process, /* CR or CR should be entered.

A sample is as follows :

```
 ENTER COMMAND NAME ===> COPY

 ENTER DSN OF DATA POOL ===> J3679.TEST00.DATA
 ENTER DSN OF 2-ND DATA POOL ===> J3679.DATAPOOL.DATA
 ENTER NODE NAME. IF *ALL IS ENTERD, ALL DATA IS COPIED
 00900 ?
NAGE.INFX.1276
 *** INFORMATION OF DATA POOL USAGE ***
     LOGICAL UNIT NO.        =     92
     DATA SET NAME           = J3679.DATAPOOL.DATA
     NO. OF WRITTEN RECORDS  =     92
     REMAINS RECORDS         =    887
 DATA COPY WAS FINISHED SUCCESSFULLY
 ENTER NEXT NODE NAME
 00900 ?
G09 .TEST
 *** INFORMATION OF DATA POOL USAGE ***
     LOGICAL UNIT NO.        =     92
     DATA SET NAME           = J3679.DATAPOOL.DATA
     NO. OF WRITTEN RECORDS  =    157
     REMAINS RECORDS         =    730
 DATA COPY WAS FINISHED SUCCESSFULLY
 ENTER NEXT NODE NAME
 00900 ?
/*
```

5.10 CONDENSE Command

The area for directories and data erased by the DELETE command remains as the unusable area, so that the release of the area should be performed by using the CONDENSE command. The condense procedure with TSS terminal or batch job can be chosen. In the case of TSS procedure, a sample is shown as follows :

```
ENTER COMMAND NAME ===> CONDENSE

EXECUTION OF CONDENSE COMMAND (TSS/BATCH) ===> TSS
ENTER DSN OF DATA POOL ===> J3679.TEST00.DATA
BACK-UP DATA SET J3679.CONDENSE          WAS CREATED
*** INFORMATION OF DATA POOL USAGE ***
    LOGICAL UNIT NO.        =     91
    DATA SET NAME           = J3679.TEST00.DATA
    NO. OF WRITTEN RECORDS  =    390
    REMAINS RECORDS         =    538
ENTRY (A) J3679.CONDENSE DELETED
```

In the case of batch job, the next Job Control Language is displayed. A TSS terminal with FSO (Full Screen Option) may be needed to use the option for batch job. The sample JCL is shown as follows :

```
 ENTER COMMAND NAME ===> COND

 EXECUTION OF CONDENSE COMMAND (TSS/BATCH) ===> BATCH

  EDIT-FSO (V01/L06) --- J3679.@POOLJCL.CNTL
==>
 ROW SCROLL ==>     PAGE      COLUMN SCROLL ==> 40          NONULLS 50
     ----*----1----*----2----*----3----*----4----*----5----*----6----*----7
 0010 //JCLG  JOB
 0020 //****************************************************************
 0030 //*        JOB CONTROL LANGUAGE FOR CONDENSE COMMAND               *
 0040 //*                                                                *
 0050 //*    PLEASE CHANGE JUSER CARD, PASSWORD, DATA POOL NAME AND      *
 0060 //*        BACK-UP FILE NAME                                       *
 0061 //*    AT END OF CHANGE PLEASE ENTER SUBMIT COMMAND                *
 0070 //*                                                                *
 0080 //****************************************************************
 0090 // EXEC JCLG
 0100 //SYSIN DD DATA,DLM='++'
 0110 // JUSER ????????,XX.XXXXXX,YYYY.ZZZ
 0120  T.4 C.1 W.0 I.5 P.0 OPN
 0130  OPTP PASSWORD=??
 0140 // EXEC LMGO,LM='J3679.POOLX',PNM=COND
 0150 //*   DATA POOL
 0160 //*     CHANGE DSN:DATA SET NAME
 0170 // EXPAND DISKTO,DDN=FT91F001,DSN='JXXXX.????????',MODE=OUT
 0180 //*   BACK-UP FILE
 0190 //*     CHANGE DSN:DATA SET NAME
 0220 // EXPAND DISKTN,DDN=FT01F001,DSN='JXXXX.@@BACKUP',UNIT=TSSWK,
 0230 //          SPC='500,300'
 0240 // EXPAND DISK,DDN=FT02F001
 0250 ++
 0260 //
 *** END OF DATA SET ***
```

The condensation starts after the backup of the data set is created, so that the recovery of the data-set can be performed by using a MTCOPY command when the condensation is abnormally terminated and the data-set is destroyed. The description of the MTCOPY command is shown in Section 5.13.

5.11 MEND Command

The MEND command is prepared for recovery of the DATA-POOL when the data-set is destroyed. The user can change data in the Control, the Directory and the comment of the Data Sections, so that the revision of tree structures, information in the sub-directory and control variables are possible. The user should search the structure of data-linkage by using the CATL command before the MEND command is executed.

At the first stage of MEND execution, 3 options can be selected by the user. The option names are CONT, DIREC and COM. This command must be entered from the first column. The CONT option is used for the revision for the Control Section of the DATA-POOL. A sample is shown as follows :

```
 ENTER COMMAND NAME ===> MEND

 ENTER DSN OF DATA POOL ===> J3679.DATAPOOL.DATA
  ++ DATA POOL INFORMATION ++
  TITLE :
          RADHEAT-V4 DATA-POOL FOR CROSS SECTIONS STORAGE
  1ST. RECORD NO. OF DIRECTORY :    2
 LAST  RECORD NO. OF DIRECTORY :    7
 REAL  RECORD NO. OF DIRECTORY :    6
 WRITE PERMIT OF THE DATA POOL :    0
 ENTER OPTION NAME  CONT/DIREC/COM/END
 00320 ?
CONT
 *********   C O N T R O L   S E C T I O N   *********
      ITEM
        1  TITLE   :
                RADHEAT-V4 DATA-POOL FOR CROSS SECTIONS STORAGE
       21  ADDRESS FOR THE DIRECTORY OF FIRST LEVEL NODE  :     2
       22  HEAD ADDRESS FOR THE VACANT DIRECTORY AREA     :     8
       23  HEAD ADDRESS FOR THE VACANT DATA AREA          :   271
       24  WRITE FLAG                                     :     1
       25  READ FLAG (NOT USED)                           :     0
       26  LENGTH OF THE ONE PHYSICAL RECORD              :   900
       27  MAXIMUM NUMBER OF THE SAME LEVEL NODE          :    74
       28  SIZE OF THE DIRECTORY SECTION                  :    20
       29  SIZE OF THE DATA SECTION                       :   979
       30  REAL NUMBER OF THE DIRECTORY RECORDS           :     6
       31  REAL NUMBER OF THE DATA SET RECORDS            :   249
 ENTER ITEM NO. TO MEND. IF ENTER 0, END TO PROCESS
 00550 ?
1
 ENTER NEW TITLE
 00630 ?
DATA-POOL COPIED FROM J3679.TEST00.DATA
 ENTER ITEM NO. TO MEND. IF ENTER 0, END TO PROCESS
 00550 ?
0
```

```
END OF MENDING A CONTROL SECTION SUCCESSFULLY
*********   C O N T R O L   S E C T I O N   *********
     ITEM
       1  TITLE    :
      DATA-POOL COPIED FROM J3679.TEST00.DATA
      21  ADDRESS FOR THE DIRECTORY OF FIRST LEVEL NODE  :    2
      22  HEAD ADDRESS FOR THE VACANT DIRECTORY AREA     :    8
      23  HEAD ADDRESS FOR THE VACANT DATA AREA          :  271
      24  WRITE FLAG                                     :    1
      25  READ FLAG (NOT USED)                           :    0
      26  LENGTH OF THE ONE PHYSICAL RECORD              :  900
      27  MAXIMUM NUMBER OF THE SAME LEVEL NODE          :   74
      28  SIZE OF THE DIRECTORY SECTION                  :   20
      29  SIZE OF THE DATA SECTION                       :  979
      30  REAL NUMBER OF THE DIRECTORY RECORDS           :    6
      31  REAL NUMBER OF THE DATA SET RECORDS            :  249
```

The user selects the item No. (1 ~ 31) to be change. The revised value or title should be entered next. These entries are given by a free format. The sequence is repeated until a CR or a 0 CR entry. The revised Control Section is displayed at the end of processing, and a next option is required. To terminate the MEND command, an END command may be entered.

The DIREC option is used to change the Directory Section. The user can change the sub-directory (SUB) and the head information defined in the directory (HEAD). A sample is shown below for the case of revision of the sub-directory.

```
 ENTER OPTION NAME  CONT/DIREC/COM/END
 00320 ?
DIREC
 ENTER NODE NAME. IF ENTER NOTHING, END TO PROCESS
 00900 ?
NAGE.INFX.1276. SMT
 ENTER OPTION NAME  SUB/LOW
 01040 ?
SUB
  ITEM          CONTENTS
     1  NODE NAME              =     SMT
     2  TOTAL LENG. OF DATA SET =      3
     3  ADDRESS OF A LOWER NODE =      0
     4  ADDRESS OF A DATA SET  =      25
     5  NO. OF SUB-DATA SETS   =       2
     6  DATE OF CREATION       =84-01-20
     8  DATA 1 =            0
     9  DATA 2 =            0
    10  DATA 3 =            0
    11  DATA 4 =            0
    12  DATA 5 =            0
 ENTER ITEM NO. TO MEND OR DEL TO DELETE THIS SUB-DIRECTORY
  IF ENTER 0, END TO MEND THE SUB-DIRECTORY
 01210 ?
11
```

```
 ENTER NEW VALUE
 01430 ?
777
 INPUT VALUE WAS INTEGER TYPE   777
 ENTER ITEM NO. TO MEND OR DEL TO DELETE THIS SUB-DIRECTORY
  IF ENTER 0, END TO MEND THE SUB-DIRECTORY
 01210 ?
0
 END OF MENDING A SUB-DIRECTORY SECTION SUCCESSFULLY
  ITEM         CONTENTS
     1  NODE NAME               =     SMT
     2  TOTAL LENG. OF DATA SET =       3
     3  ADDRESS OF A LOWER NODE =       0
     4  ADDRESS OF A DATA SET   =      25
     5  NO. OF SUB-DATA SETS    =       2
     6  DATE OF CREATION        =84-01-20
     8  DATA 1 =              0
     9  DATA 2 =              0
    10  DATA 3 =              0
    11  DATA 4 =            777
    12  DATA 5 =              0
```

In the option, the node names to be revise must be entered such as "NOD1. NOD2.NOD3.NOD4" from the first column. The user selects the item No. (1~12) to be change or a DEL command from the first column to erase the sub-directory. When the item No. is entered, the new value must be entered next. The format is free but a real quantity should be less than 13 digits contained a decimal point. The process is repeated until $\boxed{\text{CR}}$ or 0 $\boxed{\text{CR}}$is entered.

A sample is shown below when the option of "HEAD" is entered.

```
 ENTER OPTION NAME  CONT/DIREC/COM/END
 00320 ?
DIREC
 ENTER NODE NAME. IF ENTER NOTHING, END TO PROCESS
 00900 ?
NAGE.INFX.1276
 ENTER OPTION NAME  SUB/HEAD
 01040 ?
HEAD
 DIRECTORY HEAD
 ITEM     CONTENTS
   1   NODE NAME 1                1276
   2   NODE NAME 2
   3   UPPER DIRECTORY ADDRESS       4
   4   NO. OF SUB-DIRECTORY          4
 NODE NAMES FOR EACH SUB-DIRECTORY
   SMT   FTB   ELA   INS
 ENTER ITEM NO. TO MEND OR DEL TO DELETE THIS DIRECTORY HEAD
  IF ENTER 0, END TO MEND THE DIRECTORY
 01890 ?
DEL
 DIRECTORY HEAD1276 WAS DELETED
 ENTER OPTION NAME  CONT/DIREC/COM/END
 00320 ?
END
 END OF MEND COMMAND
```

The directory has the node names and the linkage information consisted of 4 variables at the first part of the each record. The user can change the tree structures of node names by using the "HEAD" option and setting 3 or 4 to the item No. If the user enters "DEL" from the first column, the directory and the linkage of the lower sub-directories is erased.

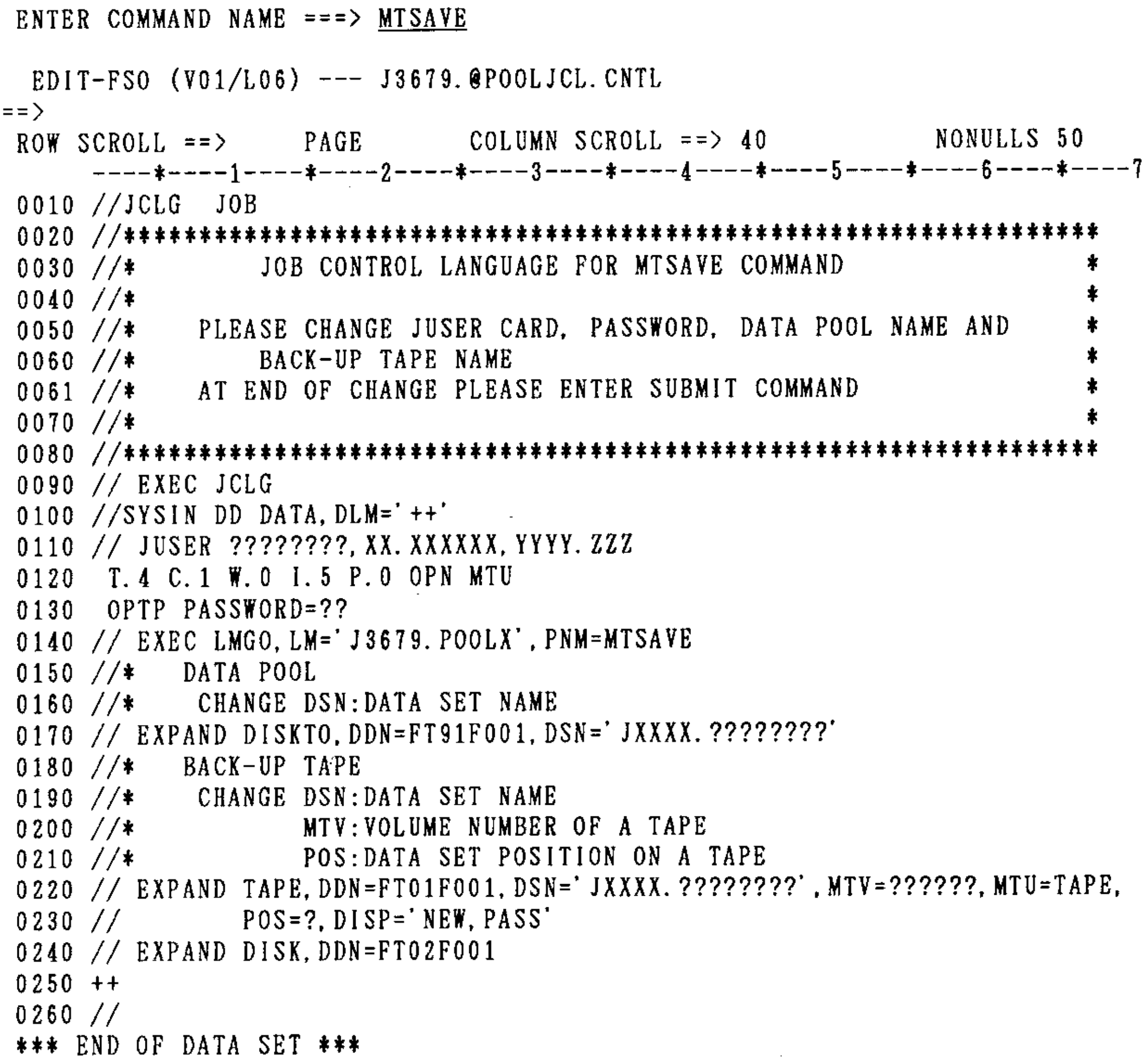
When the option of "COM" is selected, the comments of the sub-data sets are displayed. The user selects the sub-data set No. and enters a new comment (80 characters). The sequence is repeated until 0 $\boxed{\text{CR}}$ or $\boxed{\text{CR}}$ is entered. A sample is as follows :

```
 ENTER OPTION NAME  CONT/DIREC/COM/END
 00320 ?
COM
 ENTER NODE NAME. IF ENTER NOTHING. END TO PROCESS
 00900 ?
NAGE.INFX
 NO. OF SUB-DATA-SET IS    1
 DAT NO.         COMMENT
      1    INFINITE DILUTION CROSS SECTION LIBRARY
 ENTER DAT NO. TO MEND. IF ENTER 0. END TO PROCESS
 02470 ?
1
 ENTER NEW COMMENT
 02540 ?
     INFINITE DILUTION CROSS SECTIONS
 ENTER DAT NO. TO MEND. IF ENTER 0. END TO PROCESS
 02470 ?
0
 END OF MENDING DATA COMMENTS SUCCESSFULLY
 DAT NO.         COMMENT
      1       INFINITE DILUTION CROSS SECTIONS
```

The MEND command contains the complicated data entries, so that the user should take care to enter exact values. The operations flow of the MEND command is shown in Fig. 5.1.

5.12 MTSAVE Command

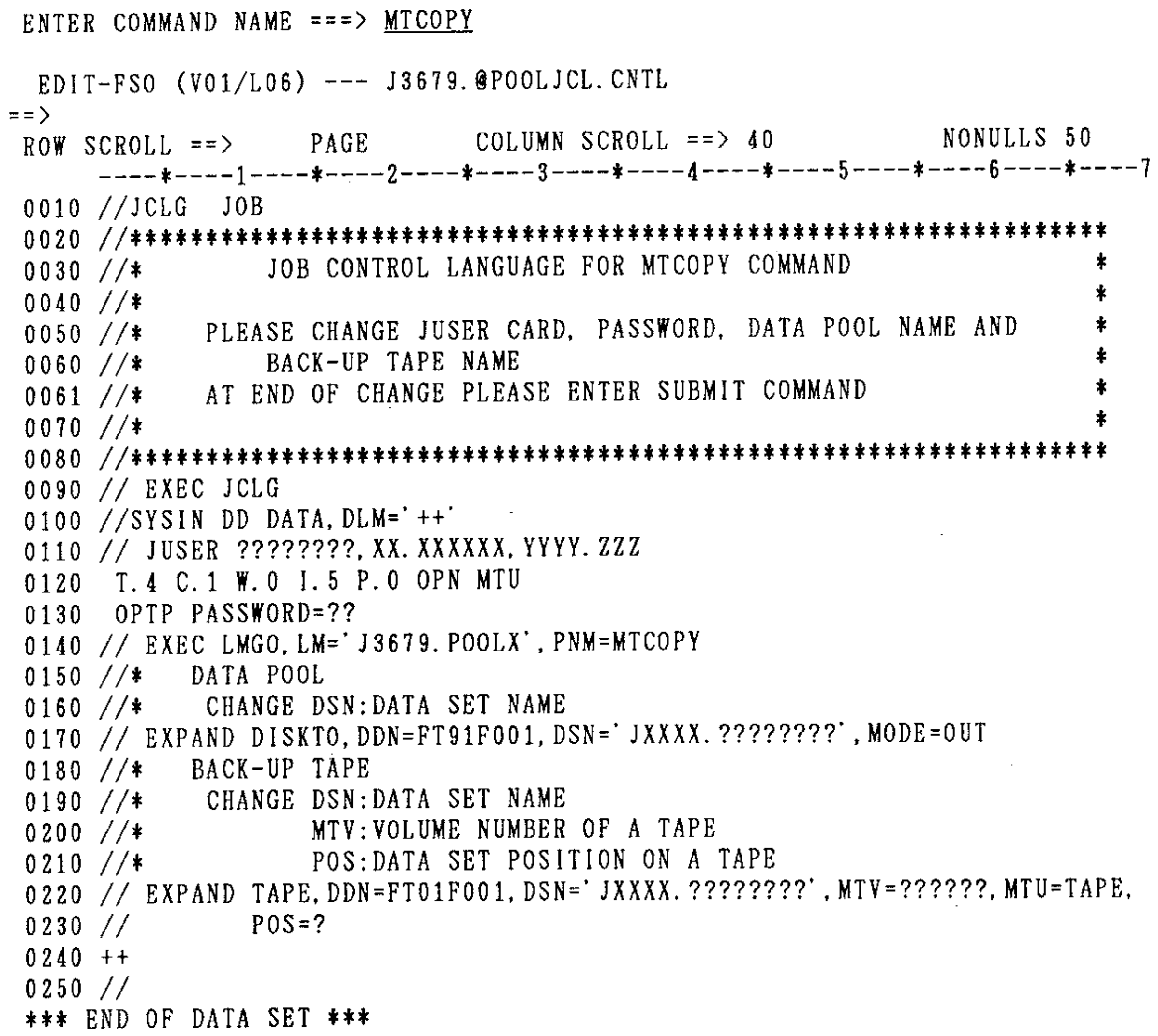
The MTSAVE command generates a Job Control Language to store data from the DATA-POOL to a backup tape or a sequential data-set. The user specifies the data-set names of the DATA-POOL and of MT, the volume serial number and the position, then enters an EDIT-mode command of "SUBMIT". A TSS terminal with FSO (Full Screen Option) may be needed to use the command.
A sample JCL is shown as follows :



```
ENTER COMMAND NAME ===> MTSAVE

 EDIT-FSO (V01/L06) --- J3679.@POOLJCL.CNTL
==>
ROW SCROLL ==>     PAGE       COLUMN SCROLL ==> 40          NONULLS 50
     ----*----1----*----2----*----3----*----4----*----5----*----6----*----7
0010 //JCLG  JOB
0020 //*******************************************************************
0030 //*         JOB CONTROL LANGUAGE FOR MTSAVE COMMAND                 *
0040 //*                                                                 *
0050 //*    PLEASE CHANGE JUSER CARD, PASSWORD, DATA POOL NAME AND       *
0060 //*        BACK-UP TAPE NAME                                        *
0061 //*    AT END OF CHANGE PLEASE ENTER SUBMIT COMMAND                 *
0070 //*                                                                 *
0080 //*******************************************************************
0090 // EXEC JCLG
0100 //SYSIN DD DATA,DLM='++'
0110 // JUSER ????????,XX.XXXXXX,YYYY.ZZZ
0120  T.4 C.1 W.0 I.5 P.0 OPN MTU
0130  OPTP PASSWORD=??
0140 // EXEC LMGO,LM='J3679.POOLX',PNM=MTSAVE
0150 //*   DATA POOL
0160 //*    CHANGE DSN:DATA SET NAME
0170 // EXPAND DISKTO,DDN=FT91F001,DSN='JXXXX.????????'
0180 //*   BACK-UP TAPE
0190 //*    CHANGE DSN:DATA SET NAME
0200 //*           MTV:VOLUME NUMBER OF A TAPE
0210 //*           POS:DATA SET POSITION ON A TAPE
0220 // EXPAND TAPE,DDN=FT01F001,DSN='JXXXX.????????',MTV=??????,MTU=TAPE,
0230 //        POS=?,DISP='NEW,PASS'
0240 // EXPAND DISK,DDN=FT02F001
0250 ++
0260 //
*** END OF DATA SET ***
```

5.13 MTCOPY Command

The MTCOPY command is prepared to recover data from a MT or a sequential data-set saved by using the MTSAVE command. The user specifies the data-set names of the DATA-POOL and of MT, the volume serial number and the position, then enters an EDIT-mode command of "SUBMIT". A sample JCL is shown as follows :



```
 ENTER COMMAND NAME ===> MTCOPY

  EDIT-FSO (V01/L06) --- J3679.@POOLJCL.CNTL
==>
 ROW SCROLL ==>     PAGE       COLUMN SCROLL ==> 40          NONULLS 50
      ----*----1----*----2----*----3----*----4----*----5----*----6----*----7
 0010 //JCLG  JOB
 0020 //**********************************************************************
 0030 //*         JOB CONTROL LANGUAGE FOR MTCOPY COMMAND                    *
 0040 //*                                                                    *
 0050 //*    PLEASE CHANGE JUSER CARD, PASSWORD, DATA POOL NAME AND          *
 0060 //*         BACK-UP TAPE NAME                                          *
 0061 //*    AT END OF CHANGE PLEASE ENTER SUBMIT COMMAND                    *
 0070 //*                                                                    *
 0080 //**********************************************************************
 0090 // EXEC JCLG
 0100 //SYSIN DD DATA,DLM='++'
 0110 // JUSER ????????,XX.XXXXXX,YYYY.ZZZ
 0120  T.4 C.1 W.0 I.5 P.0 OPN MTU
 0130  OPTP PASSWORD=??
 0140 // EXEC LMGO,LM='J3679.POOLX',PNM=MTCOPY
 0150 //*   DATA POOL
 0160 //*    CHANGE DSN:DATA SET NAME
 0170 // EXPAND DISKTO,DDN=FT91F001,DSN='JXXXX.????????',MODE=OUT
 0180 //*   BACK-UP TAPE
 0190 //*    CHANGE DSN:DATA SET NAME
 0200 //*           MTV:VOLUME NUMBER OF A TAPE
 0210 //*           POS:DATA SET POSITION ON A TAPE
 0220 // EXPAND TAPE,DDN=FT01F001,DSN='JXXXX.????????',MTV=??????,MTU=TAPE,
 0230 //        POS=?
 0240 ++
 0250 //
 *** END OF DATA SET ***
```

Note that excess data more than 50,000 records can not be treated by using the MTCOPY and the MTSAVE commands. The DATA-POOL using in the MTCOPY command should be initialized before the batch job is submitted. The title in the Control Section will be replaced to the title in the backup tape.

5.14 Operating Method of POOL for Batch Job

The load module of POOL is generated by each command name, so that the executions are performed by specifying the program name (PNM name).

```
// EXEC LMGO, LM='J3679.POOLX', PNM=program name
```

Table 5.1 shows the program names and the input/output data-sets corresponded to the command names. The input data for each program are the same as the data entries described in previous Sections. The entry of CR can be replaced by a blank card. The following cataloged procedure is useful to generate the load module of POOL.

5.15 TSS Cataloged Procedure

The cataloged procedure of POOL is shown as follows :

```
PROC 0
CONTROL PROMPT
FREE ATTRLIST(B)
FREE ATTRLIST(DCB)
ATTR B LRECL(133) RECFM(U A)
ATTR DCB LR(3600)  BL(3600) REC(F)
FREE F(FT01F001)
FREE F(FT02F001)
FREE F(FT06F001)
FREE F(FT91F001)
ALLOC DA(*) F(FT06F001) USING(B)
WRITE    ***** STARTS RADHEAT-V4 DATA POOL UTILITY *****
STAT:WRITENR ENTER COMMAND NAME ===>
READ &ELM
IF &ELM=INIT THEN GOTO JUMP1
IF &ELM=F    THEN SET &ELM=FLAG
IF &ELM=FLAG THEN GOTO JUMP2
IF &ELM=DEL  THEN SET &ELM=DELETE
IF &ELM=DELETE THEN GOTO JUMP2
IF &ELM=T    THEN SET &ELM=TREE
IF &ELM=TREE THEN GOTO JUMP2
IF &ELM=C    THEN SET &ELM=CATL
IF &ELM=CATL THEN GOTO JUMP2
IF &ELM=MEND THEN GOTO JUMP2
IF &ELM=L    THEN SET &ELM=LIST
IF &ELM=LIST THEN GOTO JUMP2
IF &ELM=RE   THEN SET &ELM=RENAME
IF &ELM=RENAME THEN GOTO JUMP2
IF &ELM=COPY THEN GOTO JUMP4
IF &ELM=CONDENSE THEN SET &ELM=COND
IF &ELM=COND THEN GOTO JUMP3
IF &ELM=MTSAVE THEN GOTO JUMP5
IF &ELM=MTCOPY THEN GOTO JUMP5
IF &ELM=HELP THEN GOTO RUN
IF &ELM=END  THEN GOTO FINIS
WRITE ERROR COMMAND NAME. PLEASE RECONFIRM BY HELP COMMAND
GOTO STAT
JUMP1:WRITENR ENTER DSN OF DATA POOL ===>
READ &DSN
WRITENR ALLOCATION OF DATA SET (NEW/OLD) =====>
READ &ANS
IF &ANS=OLD THEN +
   DO
   ALLOC F(FT91F001) DSN('&DSN') SHR
   GOTO RUN
   END
WRITENR UNIT PARAMETER         ===============>
READ &UNIT
WRITENR SPACE PARAMETER (1-ST SPACE)   =======>
READ &SPC
WRITENR SPACE PARAMETER (INCREMENT)     ======>
READ &INC
WRITENR SPACE PARAMETER (SPACE UNIT T/CY)  ===>
READ &T
IF &UNIT=MSS THEN GOTO JUMP11
ALLOC DA('&DSN') F(FT91F001) UNIT(&UNIT) SP(&SPC &INC) &T US(DCB) +
 NEW CAT
GOTO RUN
JUMP11:WRITENR MSS GROUP       ================>
READ &MSVGP
WRITENR MSS VOLUME NUMBER     ================>
READ &VOL
ALLOC DA('&DSN') F(FT91F001) UNIT(&UNIT) MSVGP(&MSVGP) VO(&VOL) +
 SP(&SPC &INC) &T US(DCB) NEW CAT
GOTO RUN
JUMP2:WRITENR ENTER DSN OF DATA POOL  ===>
READ &DSN
ALLOC F(FT91F001) DSN('&DSN') SHR
GOTO RUN
JUMP3:WRITENR EXECUTION OF CONDENSE COMMAND (TSS/BATCH) ===>
READ &TSS
```

```
IF &TSS=TSS THEN GOTO JUMP4
GOTO JUMP5
JUMP4:WRITENR ENTER DSN OF DATA POOL ===>
READ &DSN
ALLOC F(FT91F001) DSN('&DSN') SHR
IF &ELM=COPY THEN +
   DO
   WRITENR ENTER DSN OF 2-ND DATA POOL ===>
   READ &DSN2
   ALLOC DA('&DSN2') F(FT92F001) SHR
ALLOC DSN(CONDENSE) F(FT01F001) NEW SP(500 200) T UNIT(TSSWK) CAT
ALLOC F(FT02F001) NEW SP(10 10) T UNIT(TSSWK)
GOTO RUN
JUMP5:COPY 'J3679.TSSMAC.CNTL(&ELM)' @POOLJCL.CNTL
E @POOLJCL.CNTL CN;FS
DEL @POOLJCL.CNTL
GOTO STAT
RUN:CALL 'J1446.POOLX.LOAD(&ELM)'
IF &ELM NE HELP THEN FREE F(FT91F001)
IF &ELM=COPY OR &ELM=COND THEN +
   DO
   DEL CONDENSE
   FREE F(FT02F001)
   END
TOTO STAT
FINIS:FREE ATTRLIST(B)
FREE ATTRLIST(DCB)
FREE F(FT06F001)
EXIT
```

Table 5.1 Program names to execute the commands of POOL

Command Name	PNM Name	I/O Files	Comment
INIT	INIT	FT91F001	DATA-POOL
FLAG	FLAG	FT91F001	DATA-POOL
DELETE	DELETE	FT91F001	DATA-POOL
TREE	TREE	FT91F001	DATA-POOL
CATL	CATL	FT91F001	DATA-POOL
MEND	MEND	FT91F001	DATA-POOL
LIST	LIST	FT91F001	DATA-POOL
RENAME	RENAME	FT91F001	DATA-POOL
COPY	COPY	FT91F001 FT92F001 FT01F001[+1] FT02F001[+2]	DATA-POOL to be read DATA-POOL to be written working file working file
CONDENSE	COND	FT91F001 FT01F001[+1] FT02F001[+2]	DATA-POOL working file working file
MTSAVE	MTSAVE	FT91F001 FT01F001 FT02F001[+2]	DATA-POOL Magnetic Tape working file
MTCOPY	MTCOPY	FT91F001 FT01F001	DATA-POOL Magnetic Tape or Back-up data-set

[+1] This working file is used to store data in DATA-POOL, so that the sufficient space may be required, and it is desirable to allocate the working file as a cataloged data-set on TSSWK unit. When the execution of COND or COPY procedure is terminated abnormally, the data are recoverable by using the data set with MTCOPY command.

+2 This working file is used to store node names. The large space is not required.

These working files are ordinary sequential data-sets. The DCB information may be specified as LRECL=6208, BLKSIZE=6212, RECFM=VBS.

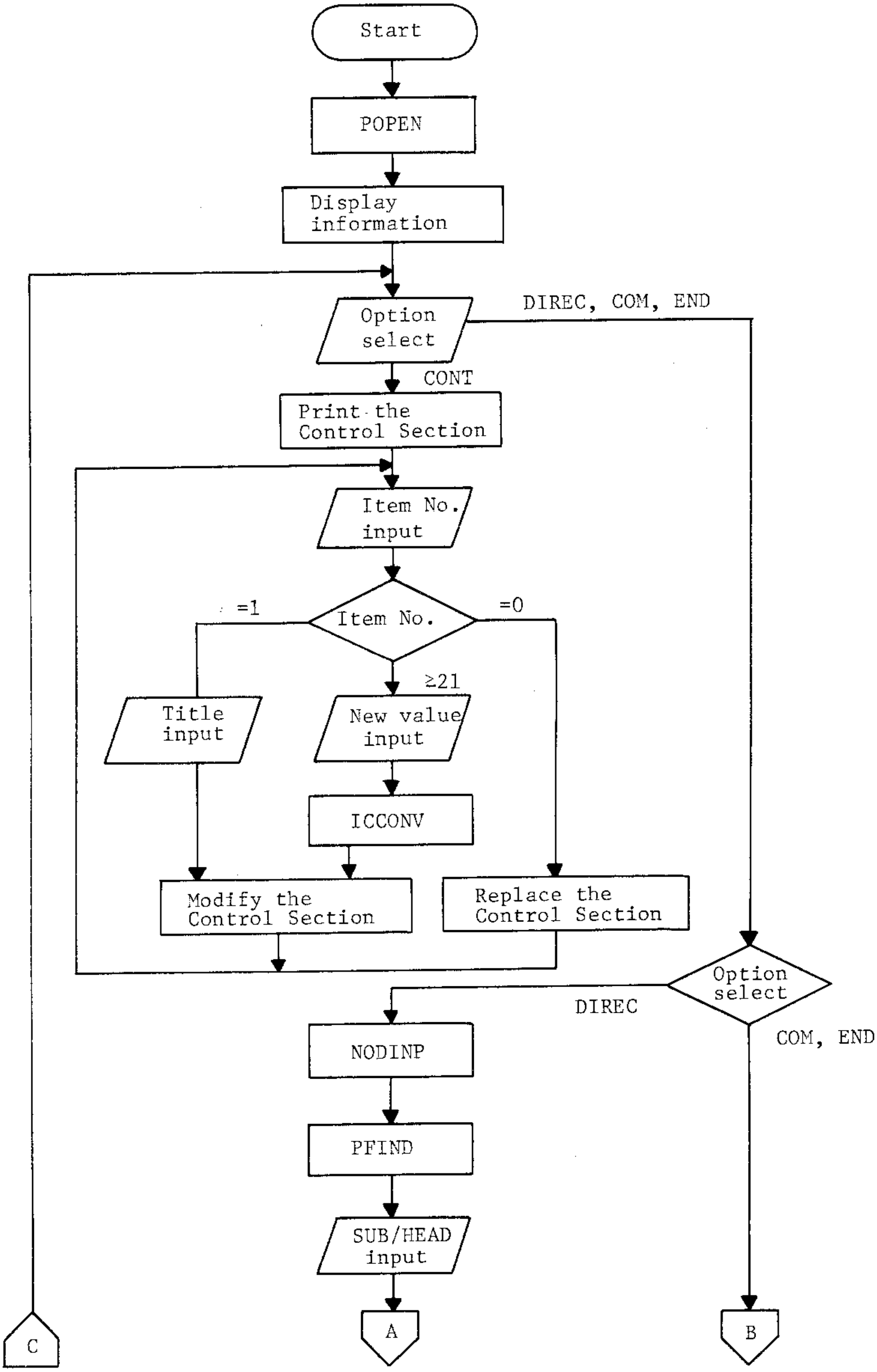


Fig. 5.1 Flow chart of MEND procedure

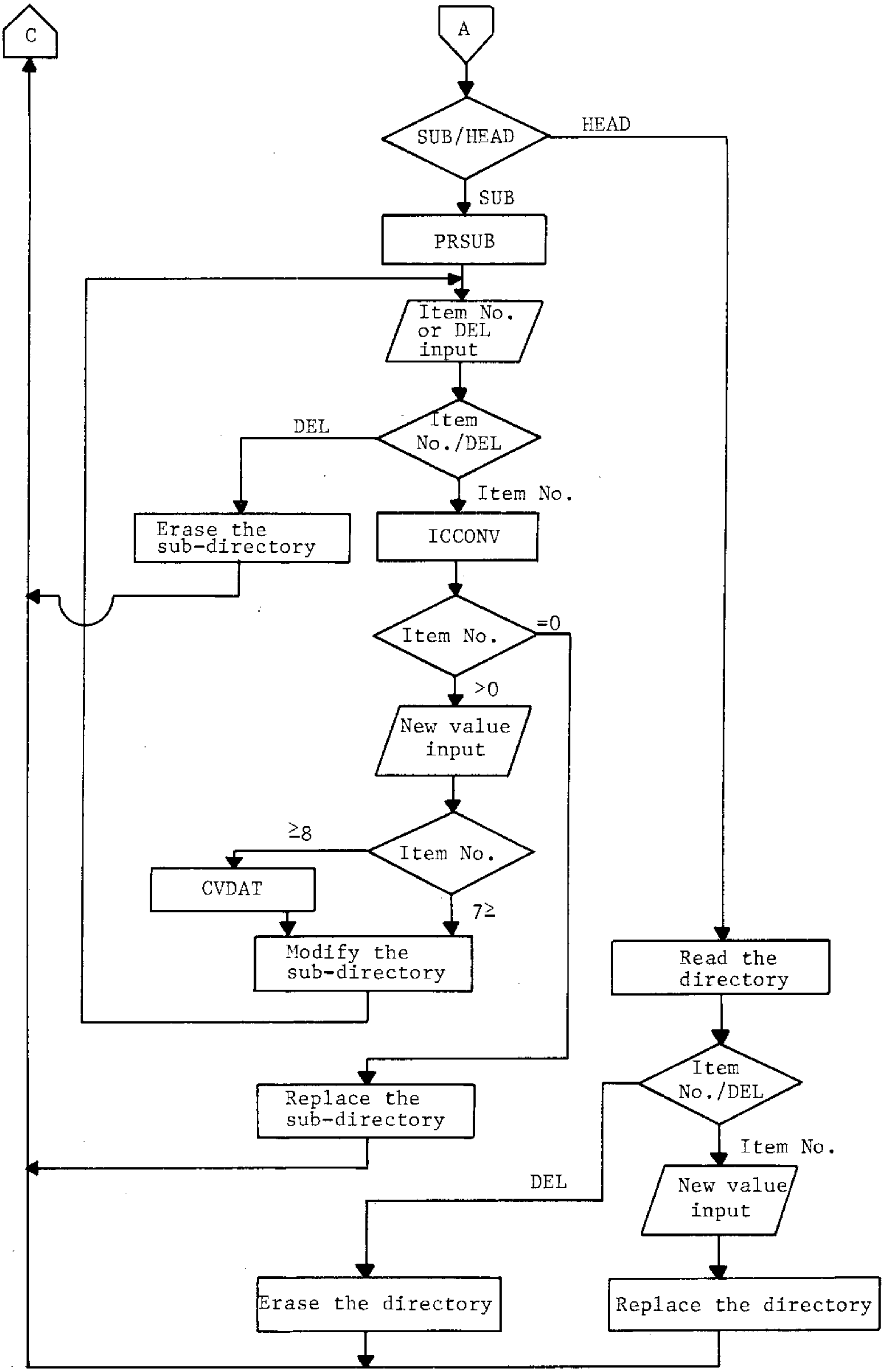


Fig. 5.1 (continued)

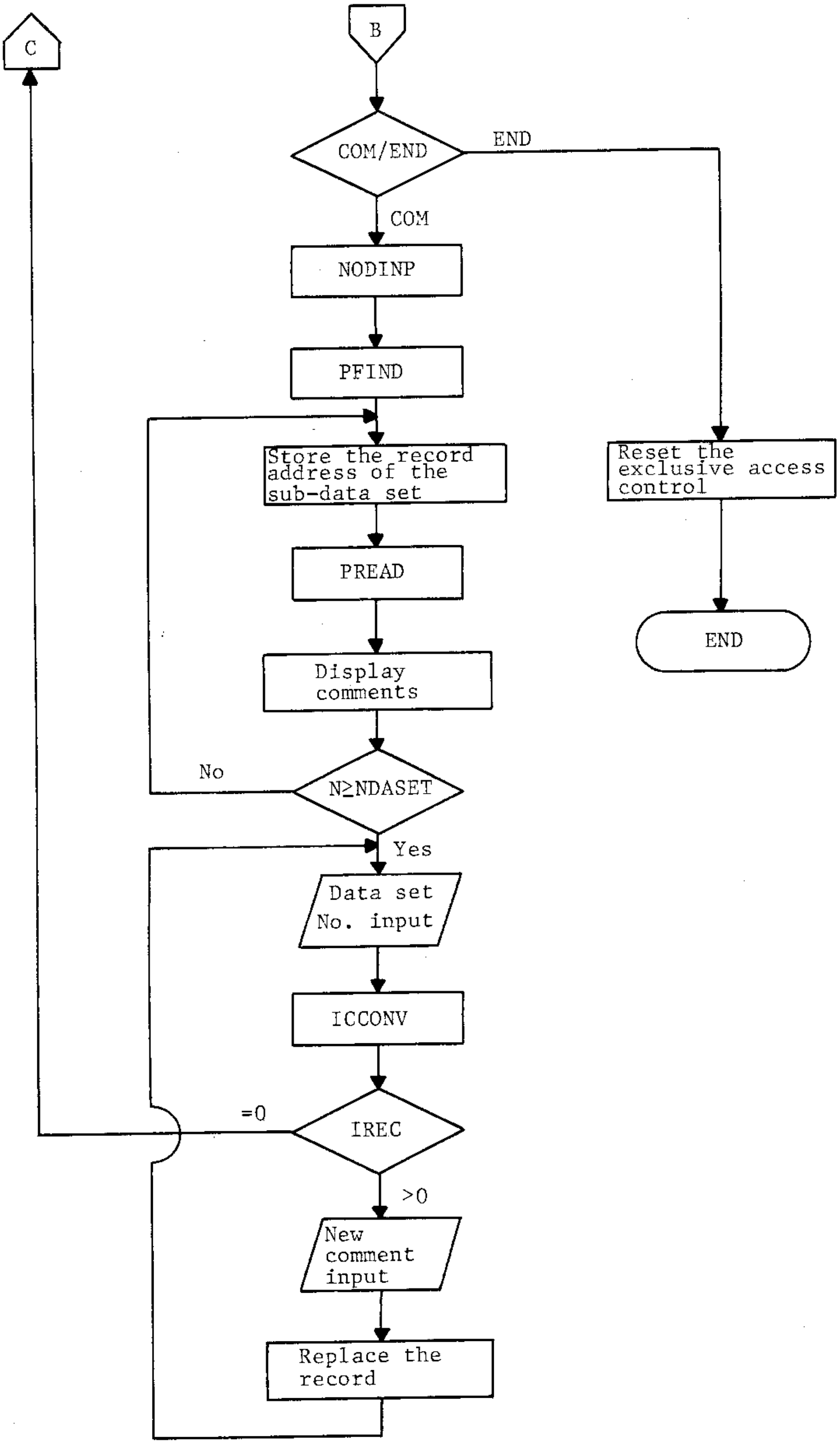


Fig. 5.1 (continued)

6. Record Format of DATA-POOL for RADHEAT-V4

The data processed in the RADHEAT-V4 code system are classified according to 8 subjects as shown in Fig. 6.1 The data in a subject relate to each other with the node tree structure. The node names and the node structure are described in this Section to display an application of the DATA-POOL for large-scale scientific computer code systems.

The data are mainly associated with the node of the last level, so that the data are classified and stored in the 17 forms described below. In the following description, the node name with a capital letter in the classification are used as the fixed name, and a small letter means that the name may change for each of data.

a) ULTX Data Form

ULTX - matno - TMPi - SIGj

This form contains the ultra-fine group cross section. The identifications for the node names are as follows :

level 1 : ULTX shows the ultra-fine group data. The energy group structure is stored in the node.

level 2 : matno shows the nuclide number. The numbers in RADHEAT-V4 correspond to those of ENDF/B-IV.

level 3 : TMPi shows the temperature. The index i indicates the temperature of i. In RADHEAT-V4, i = 1, 2, 3, 4, 5 means 300, 560, 900, 1200 and 2100 K, respectively.

level 4 : SIGj shows the background cross section. The index j indicates the σ_0 value. j = 1 means 10^8 in RADHEAT-V4.

b) SMT Data Form

EGRP -INFX - matno - SMT

This form contains the smooth cross sections with the fine-group structure. The identifications for the node names are as follows :

level 1 : EGRP shows the fine-group data. The energy group structure is stored in the node.

level 2 : INFX shows the infinitely diluted cross section.

level 3 : matno shows the nuclide number. The numbers in RADHEAT-V4 correspond to those of ENDF/B-IV.

level 4 : SMT shows the smooth cross section.

c) FTB Data Form

EGRP - INFX - matno - FTB

This form contains the self-shielding factors of each σ_0 value. The identifications for the node names from level 1 to 3 are the same as the SMT data form. The node name FTB indicates that the self-shielding facors are stored in the node.

d) ELA Data Form

EGRP -INFX - matno - ELA

This form contains the scattering matrix of elastic reaction. The identifications for the node names from level 1 to 3 are the same as the SMT data form. The node name ELA indicates that the scattering matrix of elastic reaction is stored in the node.

e) INS Data Form

EGRP -INFX - matno - INS

This form contains the scattering matrix of inelastic reaction. The identifications for the node names from level 1 to 3 are the same as the SMT data form. The node name INS indicates that the scattering matrix of inelastic reaction is stored in the node.

f) N2N Data Form

EGRP - INFX - matno - N2N

This form contains the scattering matrix of (n,2n) reaction. The identifications for the node names from level 1 to 3 are the same as the SMT data form. The node name N2N indicates that the scattering matrix of (n,2n) reaction is stored in the node.

g) H+D Data Form

EGRP - INFX - matno - H+D

This form contains the energy deposition factor and atomic displacement cross section. The identifications for the node names from level 1 to 3 are the same as the SMT data form. The node name H+D indicates that the energy deposition factor and the atomic displacement cross section are stored in the node.

h) SGRX Data Form

EGRP - SGRX - matno - ncode

This form contains the secondary gamma-ray production cross sections of each reaction. The identifications for the node names are as follows :

level 1 : EGRP shows the fine-group data same as the SMT data form.

level 2 : SGRX shows the secondary gamma-ray production cross section.

level 3 : matno shows the nuclide number. The numbers in RADHEAT-V4 correspond to those of ENDF/B-IV.

level 4 : ncode shows the reaction channel.

i) FXsn Data Form

EGRP - FXsn - matid

This form contains the effective macroscopic cross section. The identifications for the node names are as follows :

level 1 : EGRP shows the fine-group data same as the SMT data form.

level 2 : FXsn shows the effective macroscopic cross section and the number of angular points.

level 3 : matid shows the material name.

j) SELF Data Form

EGRP - SELF - matid - matno

This form contains the self-shielding factors of each nuclide in the material defined by the matid in the FXsn data form. The data are utilized for generating the effective macroscopic cross section. The identifications for the node names are as follows :

level 1 : EGRP shows the fine-group data same as the SMT data form.

level 2 : SELF shows that the self-shielding factors of each nuclide are defined in the FXsn data form.

level 3 : matid shows the material name defined in the FXsn data form.

level 4 : matno shows the nuclide number contained in the material.

k) SFX0/SFX1 Data Form

```
EGRP - problem no. ┬ SFX0
                   └ SFX1
```

This form contains forward scalar fluxes (SFX0) and adjoint scalar fluxes (SFX1) generated by an one-dimensional S_N-transport code DIAC. The data are used for generating few-group cross sections and for calculations of reaction rates. The identifications for the node names are as follows :

level 1 : EGRP shows the fine-group data same as the SMT data form.

level 2 : problem no. indicates the problem identification number specified by the first value of 15 $ array in the input data of DIAC.

level 3 : SFX0 means forward scalar fluxes of DIAC.
SFX1 shows adjoint scalar fluxes of DIAC.

l) SFX2/SFX3 Data Form

```
EGFG - problem no. ┬ SFX2
                   └ SFX3
```

This form contains forward scalar fluxes (SFX2) and adjoint scalar fluxes (SFX3) generated by a two-dimensional S_N-transport code ESPRIT. The data are used for calculations of reaction rates. The identifications for the node names are as follows :

level 1 : EGFG shows the energy group structure used in the ESPRIT calculations.

level 2 : problem no. indicates the problem identification number or name (4-characters) specified by the first data in the title card of ESPRIT.

level 3 : SFX2 shows forward scalar fluxes of ESPRIT.
SFX3 means adjoint scalar fluxes of ESPRIT.

m) AFX0/AFX1 Data Form

```
EGRP - problem no. ┬ AFX0
                   └ AFX1
```

This form contains forward angular fluxes (AFX0) and adjoint angular fluxes (AFX1) generated by the one-dimensional S_N-transport code DIAC. The identifications for the node names are as follows :

level 1 : EGRP means the energy group structure same as the SMT data form.

level 2 : problem no. is the same as the SFX0/SFX1 data form.

level 3 : AFX0 shows forward angular fluxes of DIAC.
AFX1 means adjoint angular fluxes of DIAC.

n) AFX2/AFX3 Data Form

```
EGFG - problem no. ┬ AFX2
                   └ AFX3
```

This form contains forward angular fluxes (AFX2) and adjoint angular fluxes (AFX3) generated by the two-dimensional S_N-transport code ESPRIT. The identifications for the node names are as follows :

level 1 : EGFG is the same as the SFX2/SFX3 data form.

level 2 : problem no. is the same as the SFX2/SFX3 data form.

level 3 : AFX2 shows forward angular fluxes of ESPRIT.
AFX3 means adjoint angular fluxes of ESPRIT.

o) RESD Data Form

EGRP - RESD - detector name

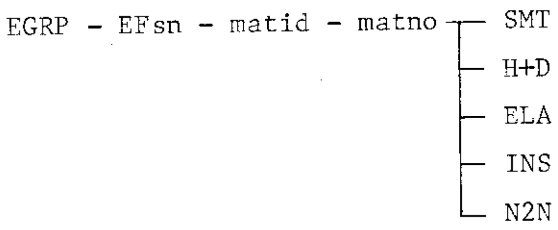
This form contains response functions to calculate reaction rates. The identifications for the node names are as follows :

level 1 : EGRP shows the energy group structure.

level 2 : RESD means the response data.

level 3 : detector name indicates identification names of the detector response functions.

p) EFsn Data Form



This form contains the effective microscopic group cross section generated by a few-group collapsing code FDEM. The identifications for the node names are as follows :

level 1 : EGRP means the energy group structure. The energy group structure is stored in the node.

level 2 : EFsn shows the effective microscopic cross section. The sn shows the number of angular meshes.

level 3 : matid shows the material identification name.

level 4 : matno shows the nuclide identification numbers contained in the material. The names are ordinarily the same as the material numbers in ENDF/B file.

level 5 : SMT, H+D, ELA, INS and N2N show reaction types of smooth cross section, energy deposition and atomic displacement, elastic scattering matrix, inelastic scattering matrix and (n,2n) scattering matrix, respectively.

g) BREM Data Form

EGRP - FXsn - matid - BREM

This form contains the Bremsstrahlung data. The secondary gamma-ray production data generated by the Bremsstrahlung effect are stored in the node. The identifications for the node names are as follows :

level 1 : EGRP shows the energy group structure. The energy group structure is contained in the node.

level 2 : FXsn shows the macroscopic cross section.

level 3 : matid shows the material identification name.

level 4 : BREM shows the Bremsstrahlung data. The secondary gamma-ray production data by the Bremsstrahlung effect.

The record formats of data generated by the RADHEAT-V4 code system are classified according to the data forms noted above and described in the following Sections. The user information of 5 words is stored by using PSET subroutine and read by using PFIND subroutine in the DATA-POOL access package, respectively. The data in the Data Section are stored by using PRITE - PRITE4 subroutines and read by using PREAD - PREAD4 subroutines. In the following description, an "information" means the user information in the Directory Section and a "data" indicates the data in the Data Section. The node name with a capital letter shows the fixed name, and a small letter means that the name changes for each data.

6.1 ULTX Data Form

```
     level 1 node :  ULTX
information NGRP, 0, 0, 0, 0
data        PREAD1 (N, NCOM, NGRP+1, FEGRP)

     level 2 node :  matno
information MATNO, MTMAX, NTMP, NSIG, LFI
data        PREAD3 (N, NCOM, MTMAX, MTYPE, NTMP, TMP, NSIG, SIGO)

     level 3 node :  TMPi
information TMP, 0, 0, 0, 0
data        PREAD (N, NCOM)

     level 4 node :  SIGj
information SIG, MTMAX2, 0, 0, 0
data        PREAD2 (N, NCOM, MTMAX2, MTYPE2, NGRP, W)
            DO 1 I = 1, MTMAX2
          1 PREAD2 (N, NCOM, 5, NDATA, M, GCS)
```

where NGRP : number of the ultra-fine energy groups,
N : logical unit number of DATA-POOL,
NCOM : comment of the node (20 words),
MATNO : material identification number,
MTMAX : number of reactions,

NTMP : number of temperatures,
NSIG : length of σ_0 table,
LFI : fission flag (0: non fission, 1: fission),
MTMAX2 : number of reactions for each σ_0 value,
FEGRP : energy group boundaries (eV),
MTYPE : reaction identification numbers,
TMP : temperatures,
SIGO : σ_0 values,
MTYPE2 : reaction identification numbers,
W : weighting spectrum,
NDATA : MTYPE(i), C1, C2, NLOW, NUP,
M : NUP-NLOW+1,
GCS : ultra-fine group cross section from the group NLOW to NUP.

6.2 SMT Data Form

```
     level 1 node : EGRP
information ING, IGG, 0, 0, 0
data        PREAD1(N, NCOM, ING+1, GNG)                (IGG=0)
            PREAD2(N, NCOM, ING+1, GNG, IGG+1, GGG)
                                                       (IGG≠0)

     level 2 node : INFX
information 0, 0, 0, 0, 0
data        PREAD (N, NCOM)

     level 3 node : matno
information MATNO, 0, 0, 0, 0
data        PREAD (N, NCOM)

     level 4 node : SMT
information 0, 0, 0, 0, 0
data        PREAD3(N, NCOM, M, MT, 1, TMP, 1, SIGO)
            PREAD1(N, NCOM, MM, SMT)
where   ING : number of neutron energy groups,
        IGG : number of gamma-ray energy groups,
          N : logical unit number of DATA-POOL,
       NCOM : comment of the node (20 words),
         MT : reaction identification numbers,
 TMP : temperature,
SIGO : σ0 value,
   M : number of reactions (10),
  MM : ING×M,
 SMT : smooth cross section.
```

6.3 FTB Data Form

```
     level 1 node : EGRP
information same as the SMT data form
data        ditto.

     level 2 node : INFX
information same as the SMT data form
data        ditto.

     level 3 node : matno
information same as the SMT data form
data        ditto.

     level 4 node : FTB
information 0, 0, 0, 0, 0
data        PREAD3(N, NCOM, M, MT, NTMP, TMP, NSIG, SIGO)
            DO 1 I=1, NTMP
          1 PREAD4(N, NCOM, LEN, SFT, LEN, SFE, LEN, SFF, LEN, SFC)
where     M : number of reactions (4),
         MT : reaction identification numbers,
       MTMP : number of temperatures,
       NSIG : number of σ0 values,
          N : logical unit number of DATA-POOL,
       NCOM : comment of the node (20 words),
        LEN : NSIG×ING,
        TMP : temperatures,
       SIGO : σ0 values,
        SFT : self-shielding factor for total reaction,
        SFE : self-shielding factor for elastic reaction,
        SFF : self-shielding factor for fission reaction,
        SFC : self-shielding factor for capture reaction.
```

6.4 ELA Data Form

level 1 node : EGRP

information same as the SMT data form

data ditto.

level 2 node : INFX

information same as the SMT data form

data ditto.

level 3 node : matno

information same as the SMT data form

data ditto.

level 4 node : ELA

information 0, 0, 0, 0, 0

data

```
       PREAD3(N, NCOM, 1, MT, 1, TMP, 1, SIGO)
       DO 1 I=1, ING, 10
     1 PREAD3(N, NCOM, ING×10, NOA, NTP, ANG, NTP, SIG)
```

where N : logical unit number of DATA-POOL,

NCOM : comment of the node (20 words),

MT : reaction identification number (MT=2),

TMP : temperature,

SIGO : σ_0 value,

NOA : number of angular points for each energy group,

NTP : summation of NOA(M) values from M=1 to M=ING×10,

ANG : cosine of scattering angles,

SIG : elastic scattering cross section in the DAR form.

6.5 INS Data Form

level 1 node : EGRP

information same as the SMT data form

data ditto.

level 2 node : INFX

information same as the SMT data form

data ditto.

level 3 node : matno

```
information same as the SMT data form
data        ditto.
```

level 4 node : INS

```
information 0, 0, 0, 0, 0
data        PREAD3(N, NCOM, 1, MT, 1, TMP, 1, SIGO)
            DO 1 I=1, ING, 10
          1 PREAD3(N, NCOM, INGx10, NOA, NTP, ANG, NTP, SIG)
where     MT : reaction identification number (MT=4),
         SIG : inelastic scattering cross section in the DAR form, the other
               notations are the same as the ELA data form.
```

6.6 N2N Data Form

level 1 node : EGRP

```
information same as the SMT data form
data        ditto.
```

level 2 node : INFX

```
information same as the SMT data form
data        ditto.
```

level 3 node : matno

```
information same as the SMT data form
data        ditto.
```

level 4 node : N2N

```
information 0, 0, 0, 0, 0
data        PREAD3(N, NCOM, 1, MT, 1, TMP, 1, SIGO)
            DO 1 I=1, ING, 10
          1 PREAD3(N, NCOM, INGx10, NOA, NTP, ANG, NTP, SIG)
where     MT : reaction identification number (MT=16),
         SIG : (n,2n) scattering cross section in the DAR form, the other
               notations are the same as the ELA data form.
```

6.7 H+D Data Form

level 1 node : EGRP

information same as the SMT data form

data ditto.

level 2 node : INFX

information same as the SMT data form

data ditto.

level 3 node : matno

information same as the SMT data form

data ditto.

level 4 node : H+D

information 0, 0, 0, 0, 0

data PREAD3(N, NCOM, M, MT, 1, TMP, 1, SIGO)
PREAD1(N, NCOM, MM, HD)

where N : logical unit number of DATA-POOL,
NCOM : comment of the node (20 words),
M : number of reaction channels (M=13),
MT : reaction identification numbers,
TMP : temperature,
SIGO : σ_0 value,
MM : ING×M,
HD : energy deposition factors and atomic displacement cross sections.

6.8 SGRX Data Form

level 1 node : EGRP

information same as the SMT data form

data ditto.

level 2 node : SGRX

information 0, 0, 0, 0, 0

data PREAD (N, NCOM)

```
     level 3 node : matno
information MATNO, 0, 0, 0, 0
data        PREAD (N, NCOM)

     level 4 node : ncode
information ITWO, ICON, KEY, NHI, NLOW
data        PREAD3(N, NCOM, LEN, X, LEN, Y, LEN1, P)
where MATNO : material identification number,
          N : logical unit number of DATA-POOL,
       NCOM : comment of the node (20 words)
       ITWO : flag of the nuclear data
              (1: ENDF/B-IV, 2: POPOP4),
       ICON : flag of the weighting procedure
              (0: constant weighting, 1: energy weighting),
        KEY : flag of the reaction
              (0: no effect, 1: inelastic excitation),
        NHI : the highest energy group for non-zero values,
       NLOW : the lowest energy group for non-zero values,
        LEN : NHI-NLOW+1,
       LEN1 : IGG×LEN,
          X : neutron interaction cross sections,
          Y : yields,
          P : probabilities ((P(i,j), i=1, IGG), j=1, LEN).
```

6.9 FXsn Data Form

```
     level 1 node : EGRP
information same as the SMT data form
data        ditto.

     level 2 node : FXsn
information IPO, 0, 0, 0, 0
data        PREAD1(N, NCOM, IPO+1, ANG)

     level 3 node : matid
information MATID, IHS, IHT, IHM, NUP
data        PREAD4(N, NCOM, NMAT, MAT1, NMAT, MAT2, NMAT, ATOM, NMAT, TMP)
            DO 1 I=1, ING+IGG
```

1 PREAD2(N, NCOM, IGT1, CRX, IGT2, CRY)

where IPO : number of fixed angular points (IPO=sn),
N : logical unit number of DATA-POOL,
NCOM : comment of the node (20 words),
MATID : material identification name,
IHS : position of self-scattering cross section,
IHT : position of total cross section,
IHM : cross section table length,
NUP : table length for up-scattering,
NMAT : number of nuclides in the material,
MAT1 : nuclide identification numbers for the SMT data,
MAT2 : nuclide identification numbers for the FTB data,
ATOM : atomic number densities (n/barn·cm),
TMP : temperatures,
LGT1 : IHM,
LGT2 : IPO×(i+NUP),
CRX : effective macroscopic cross section Σ_g,
CRY : effective macroscopic cross section $\Sigma_{g\to g',m}$.

In the data, CRX and CRY are defined by the following sequences,

Position	1 ---- NOACT+1 -- IHT IHT+1* IHT-NUP -- IHS ------------ IHM
CRX	$\Sigma_{activation}$--Σ_a $\bar{\nu}\Sigma_f$ Σ_t $\Sigma_t{}^{up}$ $\Sigma_{g+NUP\to g}$--Σ_{gg} $\Sigma_{g-1\to g}$-- $\Sigma_{1\to g}$-- 0.0

*) omit this record when NUP=0.

where NOACT is the number of the activation cross sections cosisted of the energy deposition factor and the atomic displacement cross section.

The above sequence repeats ING+IGG times.

angle No.	1	2	3		IPO
1	$\Sigma_{g+NUP\to g}(\mu_1)$	$\Sigma_{g+NUP\to g}(\mu_2)$	$\Sigma_{g+NUP\to g}(\mu_3)$	-----	$\Sigma_{g+NUP\to g}(\mu_{ipo})$
2 ⋮	⋮	⋮	⋮		⋮
NUP+1	$\Sigma_{g\to g}(\mu_1)$	$\Sigma_{g\to g}(\mu_2)$	$\Sigma_{g\to g}(\mu_3)$	---------	$\Sigma_{g\to g}(\mu_{ipo})$
NUP+2 ⋮	$\Sigma_{g-1\to g}(\mu_1)$	$\Sigma_{g-1\to g}(\mu_2)$	$\Sigma_{g-1\to g}(\mu_3)$	-------	$\Sigma_{g-1\to g}(\mu_{ipo})$
	⋮	⋮	⋮		⋮
NUP+g	$\Sigma_{1\to g}(\mu_1)$	$\Sigma_{1\to g}(\mu_2)$	$\Sigma_{1\to g}(\mu_3)$	---------	$\Sigma_{1\to g}(\mu_{ipo})$

where CRY data are stored by starting at top left corner, sweeping from left to right, then from top to bottom. The sequence repeats ING+IGG times.

6.10 SELF Data Form

level 1 node : EGRP

information same as the SMT data form

data ditto.

level 2 node : SELF

information 0, 0, 0, 0, 0

data PREAD(N, NCOM)

level 3 node : matid

information MATID, NMAT, MTMAX, 0, 0

data PREAD4(N, NCOM, NMAT, MAT1,NMAT, MAT2, NMAT, ATOM, NMAT, TMP)

level 4 node : matno

information MATNO, 0, 0, 0, 0

data PREAD4(N, NCOM, ING, FTM, ING, FEM, ING, FFM, ING, FCM)

where N : logical unit number of DATA-POOL,

NCOM : comment of the node (20 words),

MATID : material identification number,

NMAT : number of nuclides in the material,

MTMAX : number of reactions (4),

MAT1 : nuclide identification number of the SMT data,

MAT2 : nuclide identification number of the FTB data,
ATOM : atomic number densities (n/barn·cm),
TMP : temperatures,
FTM : self-shielding factor for total cross section,
FEM : self-shielding factor for elastic cross section,
FFM : self-shielding factor for fission cross section,
FCM : self-shielding factor for capture cross section.

6.11 SFX0/SFX1 Data Form

level 1 node : EGRP
information same as the SMT data form
data ditto.

level 2 node : id. name
information IGE, IM, JM, IZM, MM
data PREAD4(N, NCOM, IM+1, R, JM+1, Z, IM, MA, IZM, MZ)

where IGE : identification for geometrical configuration,
1 - slab
2 - cylinder } one-dimensional configuration,
3 - sphere
4 - (X-Y)
5 - (R-Z) } two-dimensional configuration,
6 - (R-θ)
IM : number of interval meshes for X or R axis,
JM : number of interval meshes for Y, Z or θ axis (for the case of one-dimension, JM=1),
IZM : number of zones,
MM : number of angular quadratures,
N : logical unit number of DATA-POOL,
NCOM : comment of the node (20 words),
R : spatial interval meshes for X or R axis (cm),
Z : spatial interval meshes for Y, Z or θ axis (cm),
MA : zone numbers by interval,
MZ : material numbers by interval.

level 3 node : SFX0/SFX1

SFX0 shows forward scalar flux and SFX1 means adjoint scalar flux for one-dimensional configuration.

information ING, IGG, ITH, 0, 0
data PREAD1(N, NCOM, IM×IGM, FLX)
where ITH : solution indicator (0=forward, 1=adjoint)
FLX : scalar fluxes.

6.12 SFX2/SFX3 Data Form

level 1 node : EGRP

information same as the SFX0/SFX1 data form,
data ditto.

level 2 node : id. name

information same as the SFX0/SFX1 data form,
data PREAD4(N, NCOM, IM+1, R, JM+1, Z, IM×JM, MA, IZM, MZ)
where notations are the same as those for SFX0/SFX1 data form.

level 3 node : SFX2/SFX3

SFX2 shows forward scalar flux and SFX3 means adjoint scalar flux for two-dimensional configuration.

information same as the SFX0/SFX1 data form
data DO 10 I=1, IGM
10 PREAD1(N, NCOM, IM×JM, FLX)
where notations are the same as those for SFX0/SFX1 data form.

6.13 AFX0/AFX1 Data Form

level 1 node : EGRP

information same as the SMT data form,
data ditto.

level 2 node : id. name

information same as the SFX0/SFX1 data form,
data ditto.

level 3 node : AFX0/AFX1

AFX0 shows forward angular flux and AFX1 means adjoint angular flux for two-dimensional configuration.

```
information  ING, IGG, ITH, MM, IPMESH
data         PREAD3(N, NCOM, MM, W, MM, DSN, IPMESH, NOANLL)
             DO 1 I=1, IGM
           1 PREAD1(N, NCOM, MM×IPMESH, AFX)
```

where ITH : solution indicator (0=forward, 1=adjoint),
MM : number of angular quadratures,
IPMESH : number of spatial intervals,
W : angular quadrature weights,
DSN : angular quadrature cosines,
NOANLL : spatial interval numbers,
AFX : angular fluxes.

6.14 AFX2/AFX3 Data Form

level 1 node : <u>EGRP</u>

information same as the SMT data form.
data ditto.

level 2 node : <u>id. name</u>

information same as the SFX2/SFX3 data form,
data ditto.

level 3 node : <u>AFX2/AFX3</u>

AFX2 shows forward angular flux and AFX3 means adjoint angular flux for two-dimensional configuration.

```
information  same as the AFX0/AFX1 data form
data         PREAD4(N, NCOM, MM, W, MM, AMU, MM, ETA, IPMESH, NOANLL)
             DO 1 I=1, IGM
             DO 1 J=1, IPMESH
         1   PREAD1(N, NCOM, MM×IM, AFX)
```

where W : angular quadrature weights,
AMU : angular quadrature cosines for μ,
ETA : angular quadrature cosines for η,
IPMESH : number of spatial interval meshes for Y, Z or θ axis,
AFX : angular fluxes.

6.15 RESD Data Form

level 1 node : EGRP

information same as the SMT data form,
data ditto.

level 2 node: RESD

information 0, 0, 0, 0, 0
data PREAD(N, NCOM)

level 3 node : matid

information IGM, IFLAG, 0, 0, 0
data PREAD1(N, NCOM, IGM, RD)
where IGM : number of energy groups,
IFLAG : detector identification (1 for neutron, 2 for gamma-ray),
RD : detector response function.

6.16 EFsn Data Form

level 1 node : EGRP

information same as the SMT data form,
data ditto.

level 2 node : EFsn

information IPN, 0, 0, 0, 0
data PREAD1(N, NCOM, IPN+1, ANG)
where IPN : number of angular meshes (IPN=sn),
ANG : angular meshes.

level 3 node : matid

information MATID, NMAT, 0, 0, 0
data PREAD4(N, NCOM, NMAT, MAT1, NMAT, MAT2, NMAT, ATOM, NMAT, TMP)
where notations are the same as those for the SELF data form.

level 4 node : matno

information MATNO, 0, 0, 0, 0
data PREAD (N, NCOM)

```
      level 5 node : SMT
information  0, 0, 0, 0, 0
data         PREAD3(N, NCOM, 10, MT, 1, TMP, 1, SIGO)
             PREAD1(N, NCOM, 10×INGF, CRXF)

      level 5 node : H+D
information  0, 0, 0, 0, 0
data         PREAD3(N, NCOM, 13, MT, 1, TMP, 1, SIGO)
             PREAD1(N, NCOM, 13×INGF, CRXF)

      level 5 node : ELA
information  NUPF, 0, 0, 0, 0
data         PREAD3(N, NCOM, 1, MT, 1, TMP, 1, SIGO)
             PREAD1(N, NCOM, IPN×(INGF+NUPF)×INGF, CRYF)

      level 5 node : INS
information  NUPF, 0, 0, 0, 0
data         PREAD3(N, NCOM, 1, MT, 1, TMP, 1, SIGO)
             PREAD1(N, NCOM, IPN×(INGF+NUPF)×INGF, CRYF)

      level 5 node : N2N
information  NUPF, 0, 0, 0, 0
data         PREAD3(N, NCOM, 1, MT, 1, TMP, 1, SIGO)
             PREAD1(N, NCOM, IPN×(INGF+NUPF)×INGF, CRYF)
where   MT   : reaction type identification numbers,
        TMP  : temperature (K),
        SIGO : background cross section,
        INGF : number of energy groups,
        IPN  : number of angular meshes,
        NUPF : number of up-scattering groups,
        CRXF : effective microscopic cross sections
               (the form is the same as that of CRX in the FXsn data form),
        CRYF : effective microscopic scattering matrix
               (the form is the same as that of CRY in the FXsn data form).
```

6.17 BREM Data Form

```
      level 1 node : EGRP
```

information same as the FXsn data form,
data ditto.

level 2 node : FXsn
information same as the FXsn data form,
data ditto.

level 3 node : matid
information same as the FXsn data form,
data ditto.

level 4 node : BREM
information 0, 0, 0, 0, 0
data PREAD1(N, NCOM, IGG×IGG, BR)
where IGG : number of gamma-ray energy groups,
BR(k,i): Bremsstrahlung data from group i to k.

6.18 Sample Program to Retrieve Data in DATA-POOL

To retrieve the data described in the previous Sections, the user may produces the computer program with the access package of the DATA-POOL. The sample program to obtain cross sections from the ULTX data form in the DATA-POOL is shown below. In the program, the next data are stored in one-dimensional arrays.

Item		array
The ultra-fine energy group boundaries	:	FEGRP
The identification number of reactions	:	MTYPE, MTYPE2
The temperature	:	TMP
The background cross sections	:	SIGO
The weighting spectrum	:	W
The ultra-fine group cross sections of each reaction	:	CROS

The sample program is written by FORTRAN language as follows :

```
CHARACTER    JCONTR(40),     NODE(4),      NCOM(20)
DIMENSION    FEGRP(NGRP+1),  MTYPE(MTMAX),  TMP(NTMP),  INF(12)
DIMENSION    SIGO(NSIG),  NDATA(NOARY),  MTYPE2(MTMAX2)
DIMENSION    W(NGRP),  NDATA(5),  GCS(NGRP),  CROS(NGRP, MTMAX2)

CC - DATA-POOL OPEN
     N=91
     CALL POPEN(N, JCONTR)
     NODE(1)='ULTX'
     NODE(2)='1192'
     NODE(3)='TMP1'
     NODE(4)='SIG1'
     NTH=1

CC - READ DATA OF THE FIRST LEVEL NODE
     CALL PFIND(N, NODE, NTH, INF, L)
     IF(L.NE.0) GO TO 9000
     NGRP=INF(8)
     CALL PREAD1(N, NCOM, I, FEGRP)

CC - READ DATA OF THE SECOND LEVEL NODE
     NTH=2
     CALL PFIND(N, NODE, NTH, INF, L)
     IF(L.NE.0) GO TO 9000
     MATNO=INF(8)
     MTMAX=INF(9)
     NTMP=INF(10)
     NSIG=INF(11)
     LFI=INF(12)
     CALL PREAD3(N, NCOM, MTMAX, MTYPE, NTMP, TMP, NSIG, SIGO)

CC - READ DATA OF THE THIRD LEVEL NODE
     NTH=3
     CALL PFIND(N, NODE, NTH, INF, L)
     IF(L.NE.0) GO TO 9000
     TMP=INF(8)
     CALL PREAD(N, NAME1, NAME2, NCOM, NASBD, NOSBDS, NOARY, NDATA)

CC - READ DATA OF THE FOURTH LEVEL NODE
```

```
      NTH=4
      CALL PFIND(N, NODE, NTH, INF, L)
      IF(L.NE.O) GO TO 9000
      SIG=INF(8)
      MTMAX2=INF(9)
      CALL PREAD2(N, NCOM, MTMAX2, MTYPE2, NGRP, W)
      DO 1 I=1, MTMAX2
      CALL PREAD2(N, NCOM, II, NDATA, M, GCS)
      NLOW=NDATA(4)
      NUP=NDATA(5)
      DO 3 J=1, NGRP
    3 CROS(J, I)=0.0
      K=0
      DO 2 J=NLOW, NUP
      K=K+1
    2 CROS(J, I)=GCS(K)
    1 CONTINUE
      - -
      - -
 9000 WRITE(6,9001) L
 9001 FORMAT(5X, 'THE SPECIFIED NODE IS NOT FOUND  CODE=', I5)
      STOP
```

The data defined by the other forms can be read by the same manner.

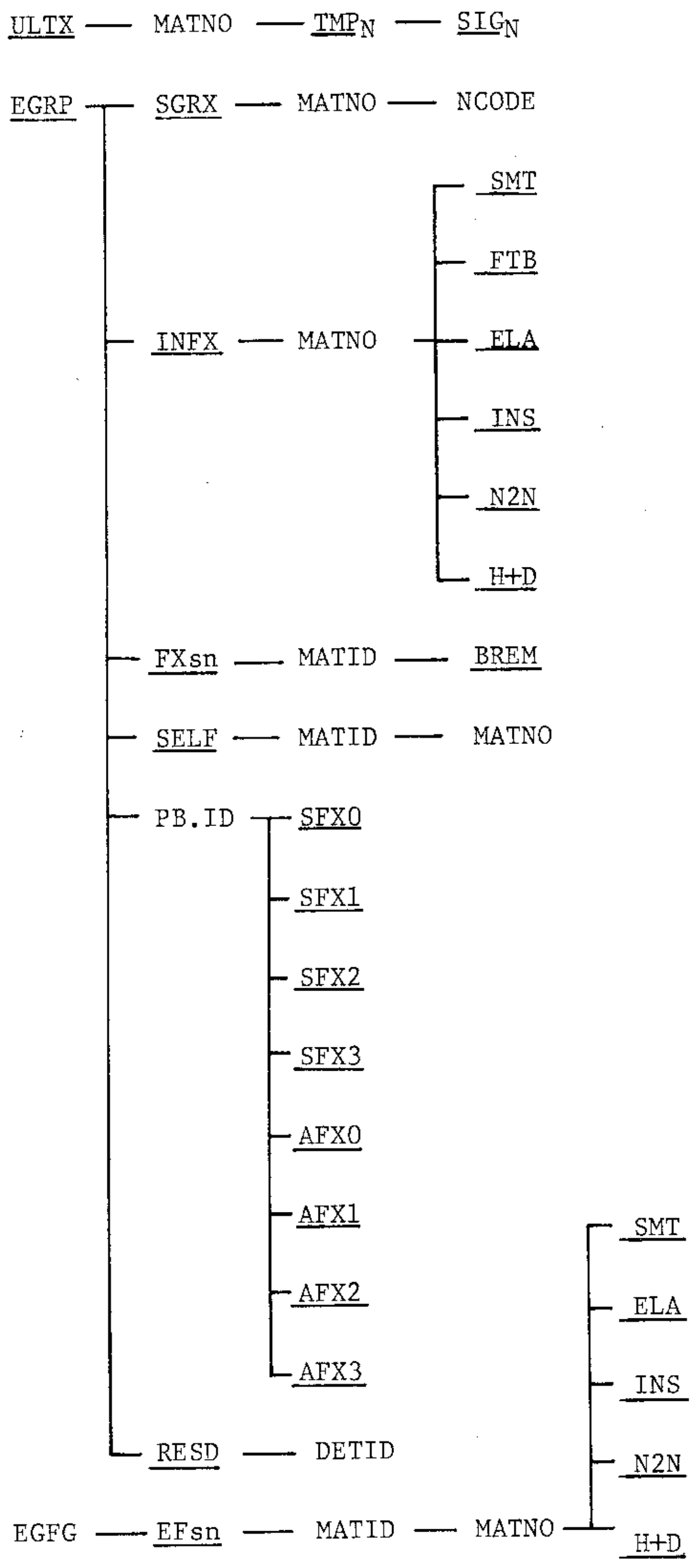


Fig. 6.1 Node tree structure adopted in the RADHEAT-V4 code system

7. Concluding Remarks

A direct-access data base DATA-POOL has been described. The access subroutines and the management utility POOL are described to utilize the DATA-POOL for large-size nuclear codes. The access package is written in the FORTRAN77 language so that the software package is applicable to the other machines.

Many samples have been shown to use the software package adequately. Error messages and program lists in Appendices may be convenient for the user to understand the access package. The code system which adopts the DATA-POOL as the standard library will be operated effectively and the data maintenance can be performed by easy operations.

Acknowledgements

Authors wish to sincere thanks to Mrs. K. Asai and M. Tomiyama for valuable discussions to develop the software package and to compare the functions with the DATAPOOL code. The comparison between the DATA-POOL access package and the DATAPOOL code is depted to Mr. T. Hara. Authors also wish to thanks to Drs. S. Katsuragi, M. Hirata and T. Asaoka for their continuous supports to our works.

References

1) Tomiyama M., Takigawa Y., Yoshimori M., Ogitsu M. and Asai K.: "Datapool; Its Concept and Facilities", JAERI-M 8715(1980).

2) Asai K., (Ed.): "Recent Code System at JAERI", JAERI-M 83-208(1983).

3) Yamano N. et al.: "RADHEAT-V4: A Code System to Generate Multigroup Constants and Analyze Radiation Transport for Shielding Safety Evaluation", JAERI-1316(1989).

4) Sasaki T. and Yamano N.: "A Software Package for Plotting Data in the RADHEAT-V4 Code System", JAERI-M 84-064 (1984).

7. Concluding Remarks

A direct-access data base DATA-POOL has been described. The access subroutines and the management utility POOL are described to utilize the DATA-POOL for large-size nuclear codes. The access package is written in the FORTRAN77 language so that the software package is applicable to the other machines.

Many samples have been shown to use the software package adequately. Error messages and program lists in Appendices may be convenient for the user to understand the access package. The code system which adopts the DATA-POOL as the standard library will be operated effectively and the data maintenance can be performed by easy operations.

Acknowledgements

Authors wish to sincere thanks to Mrs. K. Asai and M. Tomiyama for valuable discussions to develop the software package and to compare the functions with the DATAPOOL code. The comparison between the DATA-POOL access package and the DATAPOOL code is depted to Mr. T. Hara. Authors also wish to thanks to Drs. S. Katsuragi, M. Hirata and T. Asaoka for their continuous supports to our works.

References

1) Tomiyama M., Takigawa Y., Yoshimori M., Ogitsu M. and Asai K.: "Datapool; Its Concept and Facilities", JAERI-M 8715(1980).

2) Asai K., (Ed.): "Recent Code System at JAERI", JAERI-M 83-208(1983).

3) Yamano N. et al.: "RADHEAT-V4: A Code System to Generate Multigroup Constants and Analyze Radiation Transport for Shielding Safety Evaluation", JAERI-1316(1989).

4) Sasaki T. and Yamano N.: "A Software Package for Plotting Data in the RADHEAT-V4 Code System", JAERI-M 84-064 (1984).

7. Concluding Remarks

A direct-access data base DATA-POOL has been described. The access subroutines and the management utility POOL are described to utilize the DATA-POOL for large-size nuclear codes. The access package is written in the FORTRAN77 language so that the software package is applicable to the other machines.

Many samples have been shown to use the software package adequately. Error messages and program lists in Appendices may be convenient for the user to understand the access package. The code system which adopts the DATA-POOL as the standard library will be operated effectively and the data maintenance can be performed by easy operations.

Acknowledgements

Authors wish to sincere thanks to Mrs. K. Asai and M. Tomiyama for valuable discussions to develop the software package and to compare the functions with the DATAPOOL code. The comparison between the DATA-POOL access package and the DATAPOOL code is depted to Mr. T. Hara. Authors also wish to thanks to Drs. S. Katsuragi, M. Hirata and T. Asaoka for their continuous supports to our works.

References

1) Tomiyama M., Takigawa Y., Yoshimori M., Ogitsu M. and Asai K.: "Datapool; Its Concept and Facilities", JAERI-M 8715(1980).

2) Asai K., (Ed.): "Recent Code System at JAERI", JAERI-M 83-208(1983).

3) Yamano N. et al.: "RADHEAT-V4: A Code System to Generate Multigroup Constants and Analyze Radiation Transport for Shielding Safety Evaluation", JAERI-1316(1989).

4) Sasaki T. and Yamano N.: "A Software Package for Plotting Data in the RADHEAT-V4 Code System", JAERI-M 84-064 (1984).

Appendix A Error Message of DATA-POOL

Various error messages are prepared to avoid abnormal operations. The messages printed by the access subroutines are described below.

Note that the adequate message may not be displayed when logical errors exist in the user's program or the DATA-POOL is destroyed. The possible causes when the messages are printed are also described.

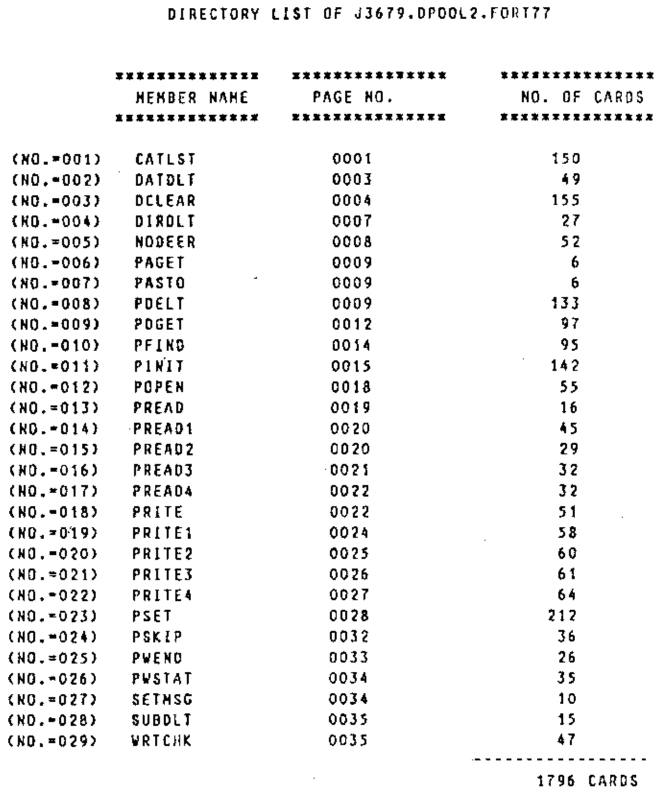
error message	subroutine	comment
ERROR CATLST THE NUMBER OF DIRECTORY IS LESS THAN ZERO	CATLST	NA1 or NA2 in the Control Section may be destroyed.
DCLEAR ERROR SUBDIRECTORY CAN NOT BE SEARCHED	DCLEAR	Node name is not found in the sub-directory. The specified node name may be mistake.
ERROR-STOP (DATA POOL ALLOCATION ERROR) LOGICAL UNIT NUMBER NODE LEVEL NODE NAME	NODEER	Error routine is called. Error is occurred at the node name of the level allocated to the logical unit number.
THIS DIRECTORY ('XXXX') DOES NOT HAVE SUBDIRECTORY	NODEER	The specified node name may be mistake or the Directory Section may be destroyed.
ALL NODE NAMES IN LEVEL X YYYY ZZZZ	NODEER	The specified node name is mistake. All node names stored in the level are printed.
LOWER NODE NAME NOT EXISTS	NODEER	The address in the sub-directory may be destroyed.
THIS NODE NAME EXISTS IN DATA-POOL ++COMPUTER MULFUNCTION++	NODEER	The node name is already exist. The Directory Section is destroyed by abnormal operations.

error message	subroutine	comment
PDELT ERROR THE SPECIFIED LEVEL OF THE NODE IS. LE. 0	PDELT	The level of the node is equal or less than zero. The specified node level may be mistake.
PDGET ERROR THE SPECIFIED LEVEL OF THE NODE IS. LE. 0	PDGET	ditto.
THE DIRECTORY OF THE NODE ('XXXX') DOES NOT HAVE SUB-DIRECTORY	PDGET	The lower nodes for the specified node name are not exist. A logical mistake may be contained in the user's program.
THE SUB-DIRECTORY OF('XXXX') IS NOT FOUND IN THE DIRECTORY OF ('YYYY')	PDGET	The Directory indicates the existence of the sub-directory, however the node name 'XXXX' is not found. The Directory Section may be destroyed.
THE DIRECTORY OF THE NODE ('XXXX') CAN NOT BE FOUND THE NODE NAME OF THE DIRECTORY IS ('YYYY')	PDGET	The Directory indicates the existence of the sub-directory, however the node name 'XXXX' is not agreed with the name in the sub-directory. The Directory Section may be destroyed.
PFIND ERROR THE SPECIFIED LEVEL OF THE NODE IS. LE. 0	PFIND	The level for the node name is equal or less than zero. The error may be occurred by a logical mistake in the user's program.
DIRECTORY DOES NOT HAVE SUB-DIRECTORY THE NODE NAME OF THE DIRECTORY IS ('XXXX')	PFIND	The error is caused by the destruction of the Directory Section.

error message	subroutine	comment
THE ADDRESS OF THE DIRECTORY IS. LE. 0 THE NODENAME OF THE DIRECTORY IS ('XXXX')	PFIND	The error is caused by the destruction of the Directory Section.
NODE NAME INPUT ERROR	PFIND	The specified node name is not found.
PINIT ERROR RECORD LENGTH IS TOO LONG. LENGTH='XXXX' MAX LENGTH='YYYY'	PINIT	A physical record length is larger than LBUFFR. The error can be erased by modifications of the access subroutines.
THE NUMBER OF DATA RECORD IS TOO LARGE UNIT='XXX' TOTAL RECORD NO='YYYYY'	PINIT	The number of records is larger than 50,000. The error can be erased by modifications of the access subroutines.
POPEN ERROR DDNAME=XXXXXXXX IS NOT ALLOCATED	POPEN	The data-set name is not found or the DCB information is mismatched.
PRITE ERROR DATA CAN NOT BE WRITTEN FOR THE LACK OF DOMAIN	PRITE	Data can not be written because the writing records will exceed the allowable limit.
PRITE1 ERROR THE SIZE OF DATA IS LESS OR EQUAL 0	PRITE1 ≀ PRITE4	The size of data to be written is equal or less than zero. The error may be caused by a logical mistake in the user's program.
PWSTAT ABORT WRITE FLAG IS ALREADY ON UNIT='XX'	PWSTAT	The exclusive access control is already operated. The latest access job may be terminated abnormally.

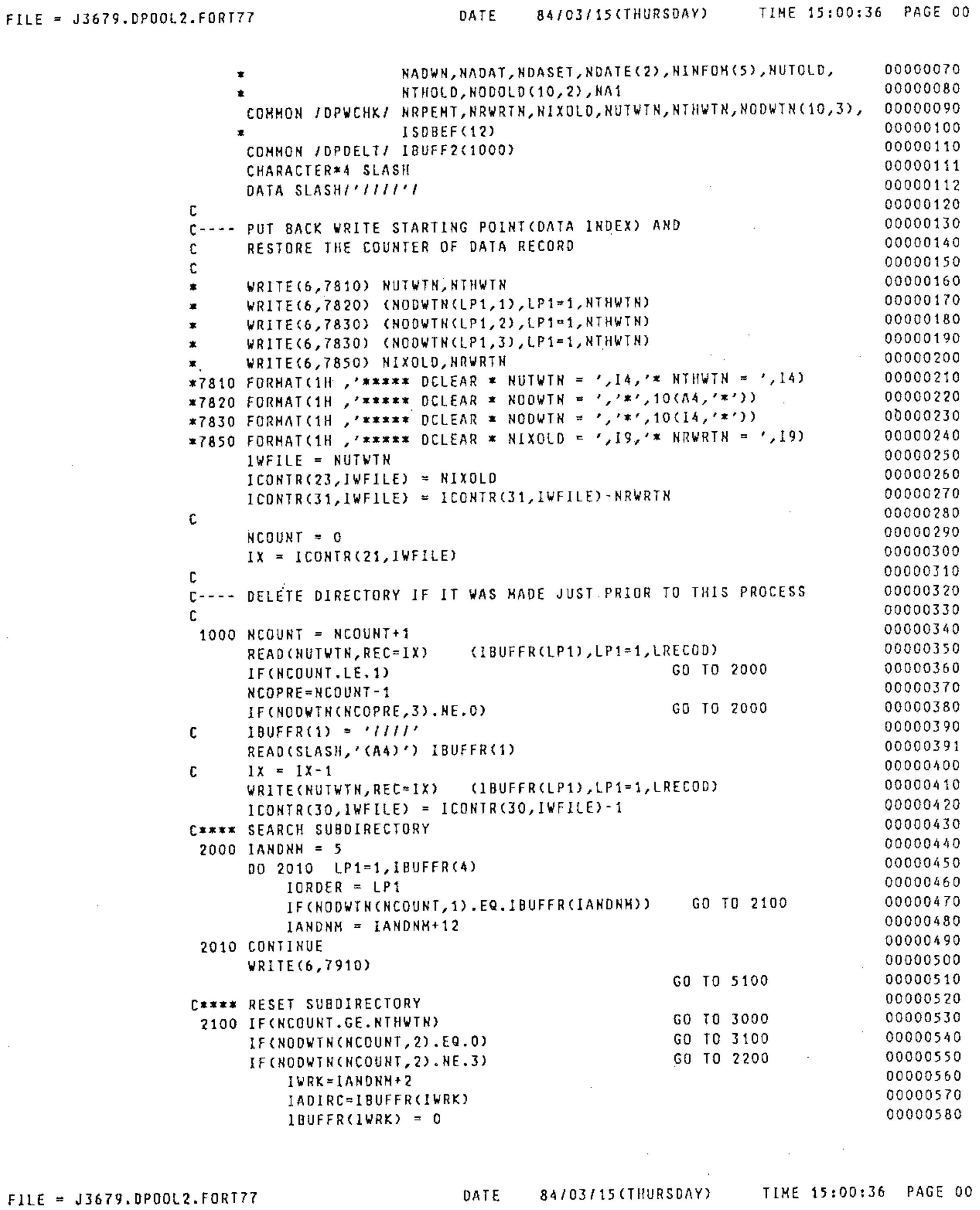
error message	subroutine	comment
PWSTAT ERROR CONTROL SECTION READ ERROR UNIT='XX'	PWSTAT	An error is occurred when the Control Section is read.
WRTCHK ERROR SUBDIRECTORY CAN NOT BE FOUND	WRTCHK	The sub-directory to be update can not be found. DATA-POOL may be destroyed.
DATA CAN NOT BE WRITTEN FOR THE LACK OF DOMAIN	PRITE1 ～ PRITE4	Data can not be written because the writing records will exceed the allowable limit. DATA-POOL is full.
PSET ERROR THE LEVEL OF THE NODE IS. LE. 0	PSET	The level of node name to be set is equal or less than zero. The error may be caused by a logical mistake in the user's program.
THE NUMBER OF SUB-DIRECTORY OF THE SAME LEVEL IS TOO LARGE NODE NAME CAN NOT BE WRITTEN	PSET	The number of nodes for the same level exceeds the allowable limit. The error can be erased by modifications of access subroutines.
DIRECTORY AREA CAN NOT BE OBTAINED	PSET	The Directory Section is full. Large directory section is needed to store the node structure.
WARNING IN PSKIP SKIPPED TO NEXT DATA SET (NODE NAME='XXXX')	PSKIP	The skipped record is the next data with the different node name.
ERROR IN PSKIP DATA SET ADDRESS WAS OVERFLOWED (IX='XXXXXXXX')		The skipped record exceeds the allowable record address.

Appendix B Program List of DATA-POOL Access Package



```
                             DIRECTORY LIST OF J3679.DPOOL2.FORT77

              **************   ***************   ***************
               MEMBER NAME       PAGE NO.          NO. OF CARDS
              **************   ***************   ***************

(NO.=001)      CATLST             0001                150
(NO.=002)      DATDLT             0003                 49
(NO.=003)      DCLEAR             0004                155
(NO.=004)      DIRDLT             0007                 27
(NO.=005)      NODEER             0008                 52
(NO.=006)      PAGET              0009                  6
(NO.=007)      PASTO              0009                  6
(NO.=008)      PDELT              0009                133
(NO.=009)      PDGET              0012                 97
(NO.=010)      PFIND              0014                 95
(NO.=011)      PINIT              0015                142
(NO.=012)      POPEN              0018                 55
(NO.=013)      PREAD              0019                 16
(NO.=014)      PREAD1             0020                 45
(NO.=015)      PREAD2             0020                 29
(NO.=016)      PREAD3             0021                 32
(NO.=017)      PREAD4             0022                 32
(NO.=018)      PRITE              0022                 51
(NO.=019)      PRITE1             0024                 58
(NO.=020)      PRITE2             0025                 60
(NO.=021)      PRITE3             0026                 61
(NO.=022)      PRITE4             0027                 64
(NO.=023)      PSET               0028                212
(NO.=024)      PSKIP              0032                 36
(NO.=025)      PWEND              0033                 26
(NO.=026)      PWSTAT             0034                 35
(NO.=027)      SETMSG             0034                 10
(NO.=028)      SUBDLT             0035                 15
(NO.=029)      VRTCHK             0035                 47
                                                 -----------------
                                                    1796 CARDS
```

```
FILE = J3679.DPOOL2.FORT77                    DATE    84/03/15(THURSDAY)     TIME 15:00:36  PAGE 00


      **************
      ** CATLST   **
      **************

                  SUBROUTINE CATLST(NUNIT)
            C 1. FUNCTION
            C  (1) PRINT OUT DIRECTORY TABLE
            C
            C---- 0. DECLARATION
            C
                  COMMON /DPCONT/ LCONTR,NCONTR,ICONTR(40,99),IX,NSUBDS,NDSTAT
                  COMMON /DPWORK/ LBUFFR,LRECOD,IBUFFR(1000),NRECOD,NODE1,NODE2,
                 *                NADWN,NADAT,NDASET,NDATE(2),NINFOM(5),NUTOLD,
                 *                NTHOLD,NODOLD(10,2),NA1
            C
                  DIMENSION IFMAT(18),IIFMAT(2),IRFMAT(2),IAFMAT(2),INFOMW(5)
                  CHARACTER IFMAT*4,IIFMAT*4,IRFMAT*4,IAFMAT*4,NCHR*1,MCHR*43,
                 *          IBLANK*4,WORD*4
            C
                  DATA IFMAT(1),IFMAT(2),IFMAT(3),IFMAT(6),IFMAT(9),IFMAT(12),
                 *     IFMAT(15),IFMAT(18)/'(1H ',' ,7X',',   ',',   ',',   ',',   '
                 *    ,',   ',')    '/
                  DATA IIFMAT(1),IIFMAT(2)/'    ',' I14'/
                  DATA IRFMAT(1),IRFMAT(2)/'1PE1','4.4 '/
                  DATA IAFMAT(1),IAFMAT(2)/'10X,','  A4'/
                  DATA IBLANK/'    '/
                  DATA MCHR/'ABCDEFGHIJKLMNOPQRSTUVWXYZ0123456789* +-=()'/
                  DATA NOCHR/43/
            C
            C---- 1. INPUT AND PRINT OUT CONTROL SECTION
            C
                  IMAX=2**30
                  IWFILE = NUNIT
                  READ(NUNIT,REC=1)      (ICONTR(LP1,IWFILE),LP1=1,LCONTR)
                  WRITE(6,7010)
                  WRITE(6,7020)
                  WRITE(6,7030) (ICONTR(LP1,IWFILE),LP1=1,18)
                  WRITE(6,7040)  ICONTR(21,IWFILE)
                  WRITE(6,7050)  ICONTR(22,IWFILE)
                  WRITE(6,7060)  ICONTR(23,IWFILE)
                  WRITE(6,7062)  ICONTR(24,IWFILE)
                  WRITE(6,7064)  ICONTR(25,IWFILE)
                  WRITE(6,7070)  ICONTR(26,IWFILE)
                  WRITE(6,7080)  ICONTR(27,IWFILE)
                  WRITE(6,7090)  ICONTR(28,IWFILE)
                  WRITE(6,7100)  ICONTR(29,IWFILE)
                  WRITE(6,7110)  ICONTR(30,IWFILE)
                  WRITE(6,7120)  ICONTR(31,IWFILE)
                  MAXLOP = ICONTR(22,IWFILE)-ICONTR(21,IWFILE)
                  IF(MAXLOP.GT.0)                                   GO TO 2010
                      WRITE(6,7910)
                                                                    GO TO 3100
            C
            C---- 2. READ AND OUTPUT DIRECTORY


FILE = J3679.DPOOL2.FORT77                    DATE    84/03/15(THURSDAY)     TIME 15:00:36  PAGE 00
FILE = J3679.DPOOL2.FORT77                    DATE     84/03/15(THURSDAY)      TIME 15:00:36  PAGE 00

C
 2010 IX = ICONTR(21,IWFILE)
      WRITE(6,7130)
C
      DO 2900  LP1 = 1,MAXLOP
          IXADRS = IX
          READ(NUNIT,REC=IX) (IBUFFR(LP2),LP2=1,LRECOD)
          IX = IX + 1
          WRITE(6,7140)  IXADRS,IBUFFR(1)
          WRITE(6,7150)  IBUFFR(3)
          WRITE(6,7160)  IBUFFR(4)
          IF(IBUFFR(4).LE.0)                           GO TO 2900
          DO 2800 LP2=1,IBUFFR(4)
               NASTAT=4+12*(LP2-1)+1
               NALAST=NASTAT-1+7
               WRITE(6,7170)
C              IF(IBUFFR(NASTAT+1).EQ.IBLANK) IBUFFR(NASTAT+1)=0
               WRITE(WORD,'(A4)') IBUFFR(NASTAT+1)
               IF(WORD.EQ.IBLANK) IBUFFR(NASTAT+1)=0
               WRITE(6,7180) LP2,(IBUFFR(LP3),LP3=NASTAT,NALAST)
               NASTAT=NASTAT+7
               NALAST=NASTAT-1+5
               NC=0
               NFADRS=4
               DO 2810 LP3=NASTAT,NALAST
                    NC=NC+1
                    INFOMW(NC)=IBUFFR(LP3)
C                 TEST IF CHARACTER
                    WRITE(NCHR,'(A1)') INFOMW(NC)
                    DO 2805 N=1,NOCHR
                         IF(NCHR.EQ.MCHR(N:N))         GO TO 2807
 2805               CONTINUE
                                                       GO TO 2806
 2807                    IFMAT(NFADRS)=IAFMAT(1)
                         NFADRS=NFADRS+1
                         IFMAT(NFADRS)=IAFMAT(2)
                                                  GO TO 2830
 2806               IWRK=INFOMW(NC)
                    IWRK=IABS(IWRK)
C                  TEST IF INTEGER
                    IF(IWRK.GE.IMAX)                   GO TO 2820
                         IFMAT(NFADRS)=IIFMAT(1)
                         NFADRS=NFADRS+1
                         IFMAT(NFADRS)=IIFMAT(2)
                                                       GO TO 2830
 2820               IFMAT(NFADRS)=IRFMAT(1)
                    NFADRS=NFADRS+1
                   IFMAT(NFADRS)=IRFMAT(2)
 2830               NFADRS=NFADRS+2
 2810          CONTINUE
               WRITE(6,7175)
               WRITE(6,IFMAT)      (INFOMW(LP3),LP3=1,5)
 2800     CONTINUE
 2900 CONTINUE
C

FILE = J3679.DPOOL2.FORT77                    DATE     84/03/15(THURSDAY)      TIME 15:00:36  PAGE 00
FILE = J3679.DPOOL2.FORT77                    DATE    84/03/15(THURSDAY)     TIME 15:00:36  PAGE 00

                C
                C----    FORMAT
                C
                 7010 FORMAT(1H1,/40X,'D I R E C T O R Y   L I S T')
                 7020 FORMAT(1H0,'*********   C O N T R O L   S E C T I O N   *********'
                     *      /1H0,5X,'COL.')
                 7030 FORMAT(1H0,4X,'1-18  TITLE   : '/1H0,6X,18A4)
                 7040 FORMAT(1H0,6X,'21  ADDRESS FOR THE DIRECTORY OF FIRST LEVEL NODE
                     *: ',I10)
                 7050 FORMAT(1H0,6X,'22  HEAD ADDRESS FOR THE VACANT DIRECTORY AREA
                     *: ',I10)
                 7060 FORMAT(1H0,6X,'23  HEAD ADDRESS FOR THE VACANT DATA AREA
                     *: ',I10)
                 7062 FORMAT(1H0,6X,'24  WRITE FLAG
                     *: ',I10)
                 7064 FORMAT(1H0,6X,'25  READ FLAG (NOT USED)
                     *: ',I10)
                 7070 FORMAT(1H0,6X,'26  LENGTH OF THE ONE PHYSICAL RECORD
                     *: ',I10)
                 7080 FORMAT(1H0,6X,'27  MAXIMUM NUMBER OF THE SAME LEVEL NODE
                     *: ',I10)
                 7090 FORMAT(1H0,6X,'28  SIZE OF THE DIRECTORY SECTION
                     *: ',I10)
                 7100 FORMAT(1H0,6X,'29  SIZE OF THE DATA SECTION
                     *: ',I10)
                 7110 FORMAT(1H0,6X,'30  REAL NUMBER OF THE DIRECTORY RECORDS
                     *: ',I10)
                 7120 FORMAT(1H0,6X,'31  REAL NUMBER OF THE DATA SET RECORDS
                     *: ',I10)
                 7910 FORMAT(1H0,5X,'***** ERROR CATLST *****',
                     */15X,'THE NUMBER OF DIRECTORY IS LESS THAN ZERO')
                 7130 FORMAT(1H0,' *********   D I R E C T O R Y   S E C T I O N   ',
                     *9('*'))
                 7140 FORMAT(1H0,2X,'*** INDEX = ',I4,' ***'/1H0,9X,'NODE NAME = ',A4)
                 7150 FORMAT(1H0,9X,'ADDRESS FOR THE UPPER NODE DIRECTORY = ',I8)
                 7160 FORMAT(1H0,9X,'NUMBER OF THE LOWER NODE             = ',I8)
                 7170 FORMAT(1H0,4X,'NO.',4X,'NODE',3X,'NRECS',5X,'NADWN',5X,'NADAT',
                     *      4X,'NDASET',5X,'DATE')
                 7175 FORMAT(1H0,14X,'INFOM(1)',6X,'INFOM(2)',6X,
                     *      'INFOM(3)',6X,'INFOM(4)',6X,'INFOM(5)')
                 7180 FORMAT(1H ,3X,I4,4X,A4,4X,I4,3(2X,I8),2X,2A4)
                C
                      RETURN
                 3100 STOP
                      END

      **************
      ** DATDLT   **
      **************

                      SUBROUTINE DATDLT(NUNIT,NSDOLD)
                C
                C---- DECLARATION
                C
                      COMMON /DPCONT/ LCONTR,NCONTR,ICONTR(40,99),IX,NSUBDS,NDSTAT

FILE = J3679.DPOOL2.FORT77                    DATE    84/03/15(THURSDAY)     TIME 15:00:36  PAGE 00
FILE = J3679.DPOOL2.FORT77                   DATE    84/03/15(THURSDAY)    TIME 15:00:36  PAGE 00

            COMMON /DPWORK/ LBUFFR,LRECOD,IBUFFR(1000),NRECOD,NODE1,NODE2,     00000060
           *                NADWN,NADAT,NDASET,NDATE(2),NINFOM(5),NUTOLD,      0.0000070
           *                NTHOLD,NODOLD(10,2),NA1                            00000080
            CHARACTER*4 SLASH                                                  00000081
            DATA SLASH/'////'/                                                 00000082
      C                                                                        00000090
            DIMENSION NSDOLD(12)                                               00000100
      C                                                                        00000110
      C---- DELETE DATA                                                        00000120
      C                                                                        00000130
            IF(NSDOLD(4).LE.0)                          GO TO 2000             00000140
            IWFILE = NUNIT                                                     00000150
            IX = NSDOLD(4)                                                     00000160
            NDTOTL = 0                                                         00000170
      C                                                                        00000180
            IF(NSDOLD(5).LE.0)                          GO TO 2000             00000190
            DO 1010  LP1=1,NSDOLD(5)                                           00000200
                READ(NUNIT,REC=IX) (IBUFFR(LP2),LP2=1,LRECOD)                  00000210
      C         IBUFFR(1) = '////'                                             00000220
                READ(SLASH,'(A4)') IBUFFR(1)                                   00000221
      C         IX = IX-1                                                      00000230
                WRITE(NUNIT,REC=IX) (IBUFFR(LP2),LP2=1,LRECOD)                 00000240
                NWSIZE=0                                                       00000250
                IF(IBUFFR(25).LE.0)                     GO TO 1210             00000260
                     DO 1110   LP2=1,IBUFFR(25)                                00000270
                          IWRK=25+LP2                                          00000280
                          NWSIZE=NWSIZE+IBUFFR(IWRK)                           00000290
       1110          CONTINUE                                                  00000300
       1210     CONTINUE                                                       00000310
                NWSIZE=NWSIZE+25+IBUFFR(25)                                    00000320
                MAXLOP=NWSIZE/LRECOD                                           00000330
                IWRK=MOD(NWSIZE,LRECOD)                                        00000340
      C                                                                        00000350
                IF(IWRK.EQ.0)                           GO TO 1220             00000360
                     MAXLOP=MAXLOP+1                                           00000370
      C1220     IX=IX-1+MAXLOP                                                 00000380
       1220     IX=IX+MAXLOP                                                   00000381
                NDTOTL = NDTOTL+MAXLOP                                         00000390
      C                                                                        00000400
       1010 CONTINUE                                                           00000410
            ICONTR(31,IWFILE)=ICONTR(31,IWFILE)-NDTOTL                         00000420
      C                                                                        00000430
       2000 RETURN                                                             00000440
            END                                                                00000450

**************
** DCLEAR   **
**************

            SUBROUTINE DCLEAR                                                  00000010
      C                                                                        00000020
      C---- DECLARATION                                                        00000030
      C                                                                        00000040
            COMMON /DPCONT/ LCONTR,NCONTR,ICONTR(40,99),IX,NSUBDS,NDSTAT       00000050
            COMMON /DPWORK/ LBUFFR,LRECOD,IBUFFR(1000),NRECOD,NODE1,NODE2,     00000060


FILE = J3679.DPOOL2.FORT77                   DATE    84/03/15(THURSDAY)    TIME 15:00:36  PAGE 00
```



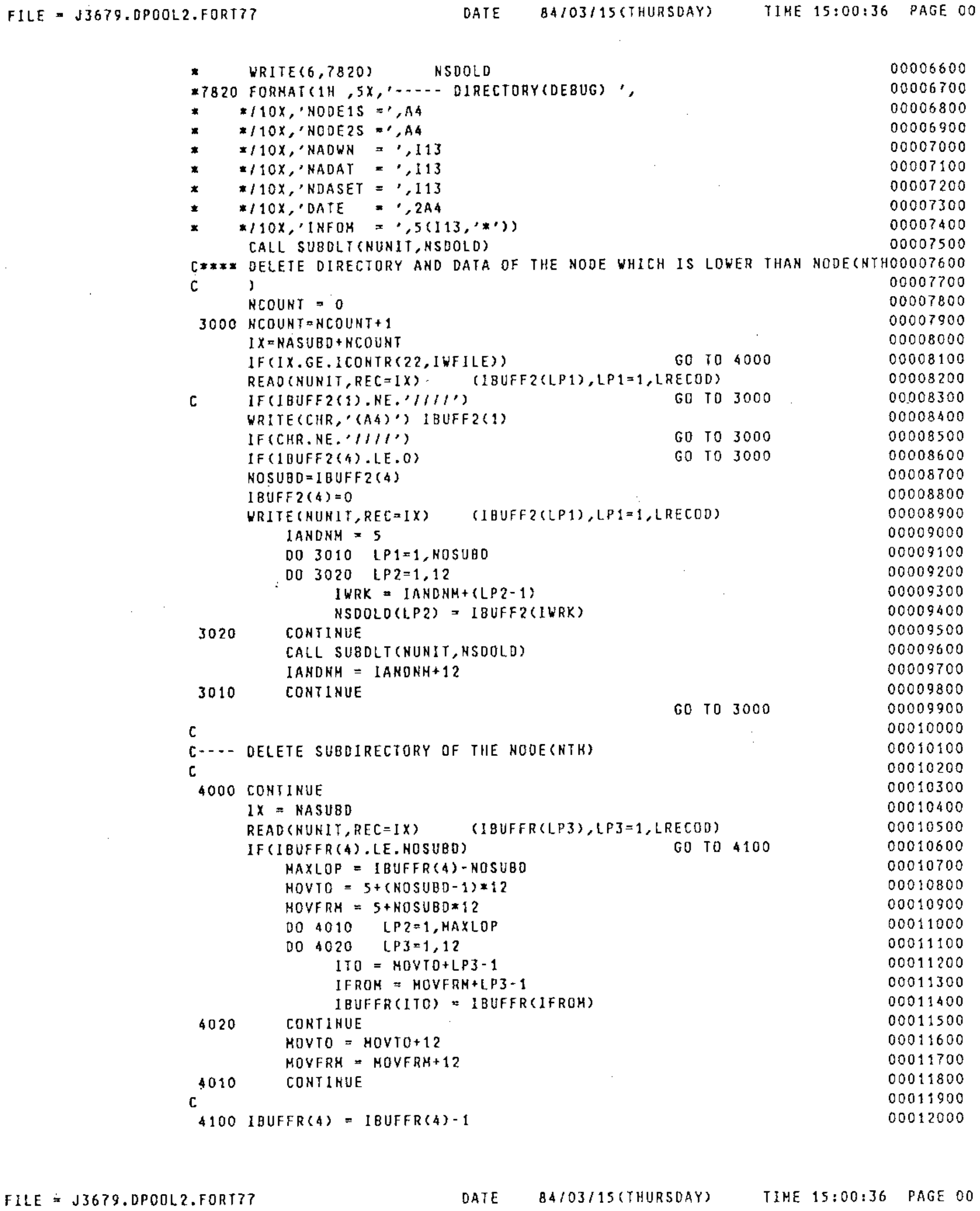
```
FILE = J3679.DPOOL2.FORT77                 DATE    84/03/15(THURSDAY)     TIME 15:00:36  PAGE 00

     *                NADWN,NADAT,NDASET,NDATE(2),NINFOM(5),NUTOLD,
     *                NTHOLD,NODOLD(10,2),NA1
      COMMON /DPWCHK/ NRPEMT,NRWRTN,NIXOLD,NUTWTN,NTHWTN,NODWTN(10,3),
     *                ISDBEF(12)
      COMMON /DPDELT/ IBUFF2(1000)
      CHARACTER*4 SLASH
      DATA SLASH/'////'/
C
C---- PUT BACK WRITE STARTING POINT(DATA INDEX) AND
C     RESTORE THE COUNTER OF DATA RECORD
C
*     WRITE(6,7810) NUTWTN,NTHWTN
*     WRITE(6,7820) (NODWTN(LP1,1),LP1=1,NTHWTN)
*     WRITE(6,7830) (NODWTN(LP1,2),LP1=1,NTHWTN)
*     WRITE(6,7830) (NODWTN(LP1,3),LP1=1,NTHWTN)
*     WRITE(6,7850) NIXOLD,NRWRTN
*7810 FORMAT(1H ,'***** DCLEAR * NUTWTN = ',I4,'* NTHWTN = ',I4)
*7820 FORMAT(1H ,'***** DCLEAR * NODWTN = ','*',10(A4,'*'))
*7830 FORMAT(1H ,'***** DCLEAR * NODWTN = ','*',10(I4,'*'))
*7850 FORMAT(1H ,'***** DCLEAR * NIXOLD = ',I9,'* NRWRTN = ',I9)
      IWFILE = NUTWTN
      ICONTR(23,IWFILE) = NIXOLD
      ICONTR(31,IWFILE) = ICONTR(31,IWFILE)-NRWRTN
C
      NCOUNT = 0
      IX = ICONTR(21,IWFILE)
C
C---- DELETE DIRECTORY IF IT WAS MADE JUST PRIOR TO THIS PROCESS
C
 1000 NCOUNT = NCOUNT+1
      READ(NUTWTN,REC=IX)    (IBUFFR(LP1),LP1=1,LRECOD)
      IF(NCOUNT.LE.1)                              GO TO 2000
      NCOPRE=NCOUNT-1
      IF(NODWTN(NCOPRE,3).NE.0)                    GO TO 2000
C     IBUFFR(1) = '////'
      READ(SLASH,'(A4)') IBUFFR(1)
C     IX = IX-1
      WRITE(NUTWTN,REC=IX)    (IBUFFR(LP1),LP1=1,LRECOD)
      ICONTR(30,IWFILE) = ICONTR(30,IWFILE)-1
C**** SEARCH SUBDIRECTORY
 2000 IANDNM = 5
      DO 2010  LP1=1,IBUFFR(4)
          IORDER = LP1
          IF(NODWTN(NCOUNT,1).EQ.IBUFFR(IANDNM))     GO TO 2100
          IANDNM = IANDNM+12
 2010 CONTINUE
      WRITE(6,7910)
                                                   GO TO 5100
C**** RESET SUBDIRECTORY
 2100 IF(NCOUNT.GE.NTHWTN)                         GO TO 3000
      IF(NODWTN(NCOUNT,2).EQ.0)                    GO TO 3100
      IF(NODWTN(NCOUNT,2).NE.3)                    GO TO 2200
          IWRK=IANDNM+2
          IADIRC=IBUFFR(IWRK)
          IBUFFR(IWRK) = 0

FILE = J3679.DPOOL2.FORT77                 DATE    84/03/15(THURSDAY)     TIME 15:00:36  PAGE 00
FILE = J3679.DPOOL2.FORT77                DATE    84/03/15(THURSDAY)     TIME 15:00:36  PAGE 00

C         IX = IX-1
          WRITE(NUTWTN,REC=IX)    (IBUFFR(LP1),LP1=1,LRECOD)
                                                 GO TO 2300
C**** CASE DIRECTORY IS NOT CHANGED
 2200 IWRK = IANDNM+2
      IADIRC = IBUFFR(IWRK)
C**** SET LOWER DIRECTORY ADRRESS
 2300 IX = IADIRC
      GO TO 1000
C
C---- SUBDIRECTORY OF THE NODE NAME THAT IS LAST OR OVERFLOWED
C
 3000 IF(NODWTN(NCOUNT,2).NE.0)                  GO TO 4000
C**** NODWTN(NCOUNT,2) IS ZERO. DELETE SUBDIRECTORY
 3100 IWRK=IANDNM+2
      IADIRC=IBUFFR(IWRK)
      IOTHER=IBUFFR(4)-IORDER
      IF(IOTHER.LE.0)                            GO TO 3200
          IWRK = IANDNM-1
          DO 3120 LP1=1,IOTHER
          DO 3130 LP2=1,12
               IWRK=IWRK+1
               IWRK1=IWRK+12
               IBUFFR(IWRK) = IBUFFR(IWRK1)
 3130     CONTINUE
 3120     CONTINUE
C
 3200 IBUFFR(4)=IBUFFR(4)-1
C         IX=IX-1
      WRITE(NUTWTN,REC=IX)   (IBUFFR(LP1),LP1=1,LRECOD)
      IF(NCOUNT.GE.NTHWTN)                          GO TO 5200
          IX=IADIRC
                                                 GO TO 1000
C**** NODWTN(NCOUNT,2) IS 4
C**** RESTORE SUBDIRECTORY AND LINK OLD DATA TO SUBDIRECTORY
 4000 IF(NODWTN(NCOUNT,2).NE.4)                       GO TO 4500
      IWRK=IANDNM-1
      DO 4010 LP1=1,12
          IWRK=IWRK+1
          IBUFFR(IWRK)=ISDBEF(LP1)
 4010 CONTINUE
      DO 4020 LP1=1,LRECOD
          IBUFF2(LP1)=IBUFFR(LP1)
 4020 CONTINUE
C     IXOLD=IX-1
      IXOLD=IX
      IF(ISDBEF(4).EQ.0)                              GO TO 4400
      IX=ISDBEF(4)
      NDTOTL = 0
      DO  4110 LP1=1,ISDBEF(5)
          READ(NUTWTN,REC=IX)    (IBUFFR(LP2),LP2=1,LRECOD)
          IBUFFR(1)=ISDBEF(1)
C         IX=IX-1
          WRITE(NUTWTN,REC=IX)    (IBUFFR(LP2),LP2=1,LRECOD)
          NWSIZE=0

FILE = J3679.DPOOL2.FORT77                DATE    84/03/15(THURSDAY)     TIME 15:00:36  PAGE 00
FILE = J3679.DPOOL2.FORT77                 DATE    84/03/15(THURSDAY)     TIME 15:00:36  PAGE 00

                IF(IBUFFR(25).LE.0)                          GO TO 4200
                DO 4120   LP2=1,IBUFFR(25)
                  IWRK=25+LP2
                  NWSIZE=NWSIZE+IBUFFR(IWRK)
         4120   CONTINUE
         4200   NWSIZE=NWSIZE+25+IBUFFR(25)
                MAXLOP=NWSIZE/LRECOD
                IWRK=MOD(NWSIZE,LRECOD)
                IF(IWRK.EQ.0)                                GO TO 4300
                  MAXLOP=MAXLOP+1
        C4300   IX=IX-1+MAXLOP
         4300   IX=IX+MAXLOP
                NDTOTL=NDTOTL+MAXLOP
         4110 CONTINUE
              ICONTR(31,IWFILE)=ICONTR(31,IWFILE)+NDTOTL
         4400 IX=IXOLD
              WRITE(NUTWTN,REC=IX)   (IBUFF2(LP1),LP1=1,LRECOD)
                                                                  GO TO 5200
        C**** NODWTN(NCOUNT,2) IS 3
        C**** RESET ADRRESS FOR THE DIRECTORY OF THIS LOWER NODE
         4500 IF(NODWTN(NCOUNT,2).NE.3)                           GO TO 5200
              IWRK=IANDNM+2
              IADIRC=IBUFFR(IWRK)
              IBUFFR(IWRK)=0
        C     IX=IX-1
              WRITE(NUTWTN,REC=IX)   (IBUFFR(LP1),LP1=1,LRECOD)
        C
        C---- RETURN
        C
         5200 RETURN
         5100 ICONTR(24,IWFILE)=0
              WRITE(NUTWTN,REC=1)    (ICONTR(LP2,IWFILE),LP2=1,LCONTR)
              STOP
        C
        C---- FORMAT
        C
         7910 FORMAT(1H ,'***** DCLEAR ERROR *****'
             *15X,'SUBDIRECTORY CAN NOT BE SEARCHED')
              END

**************
** DIRDLT   **
**************

              SUBROUTINE DIROLT(NUNIT,NSDOLD)
        C
        C---- DECLARATION
        C
              COMMON /DPWORK/ LBUFFR,LRECOD,IBUFFR(1000),NRECOD,NODE1,NODE2,
             *                NADWN,NADAT,NDASET,NDATE(2),NINFON(5),NUTOLD,
             *                NTHOLD,NODOLD(10,2),NA1
              COMMON /DPCONT/ LCONTR,NCONTR,ICONTR(40,99),IX,NSUBDS,NDSTAT
              CHARACTER*4 SLASH
              DATA SLASH/'/////'/
        C

FILE = J3679.DPOOL2.FORT77                 DATE    84/03/15(THURSDAY)     TIME 15:00:36  PAGE 00
```

```
      DIMENSION NSDOLD(1)
C
C---- DELETE DIRECTORY (SET '////' INTO NODE NAME OF DIRECTORY)
C
      IF(NSDOLD(3).LE.0)                      GO TO 2000
      IX = NSDOLD(3)
      READ(NUNIT,REC=IX)      (IBUFFR(LP1),LP1=1,LRECOD)
C     IBUFFR(1) = '////'
      READ(SLASH,'(A4)') IBUFFR(1)
C     IX = IX-1
      WRITE(NUNIT,REC=IX)     (IBUFFR(LP1),LP1=1,LRECOD)
      IWFILE = NUNIT
      ICONTR(30,IWFILE) = ICONTR(30,IWFILE)-1
C
 2000 RETURN
      END
```

```
**************
** NODEER   **
**************
```

```
      SUBROUTINE  NODEER(NUNIT,NTH,NODE)
C
C---  DECLARATION
C
      COMMON /DPCONT/ LCONTR,NCONTR,ICONTR(40,99),IX,NSUBDS,NDSTAT
      COMMON /DPWORK/ LBUFFR,LRECOD,IBUFFR(1000),NRECOD,NODE1,NODE2,
     *                NADWN,NADAT,NDASET,NDATE(2),NINFOM(5),NUTOLD,
     *                NTHOLD,NODOLD(10,2),NA1
C
      DIMENSION NODE(1),MODE(100)
C
C---  ERROR ROUTINE  ( OUTPUT NODE NAME AND DEPTH )
C
      WRITE(6,9000)  NUNIT,NTH,(NODE(N),N=1,NTH)
      NCOUNT = 0
      IWFILE = NUNIT
      IX     = ICONTR(21,IWFILE)
 1000 NCOUNT = NCOUNT+1
      READ(NUNIT,REC=IX) (IBUFFR(LP1),LP1=1,LRECOD)
      IF(IBUFFR(4).GT.0)    GO TO 1010
      WRITE(6,9010)  IBUFFR(1)
      STOP
 1010 IANDNM = 5
      NBUF4  = IBUFFR(4)
      DO 1020 LP1=1,NBUF4
      MODE(LP1)=IBUFFR(IANDNM)
      IF(NODE(NCOUNT).EQ.IBUFFR(IANDNM))  GO TO 1030
      IANDNM = IANDNM+12
 1020 CONTINUE
      WRITE(6,9020)  NCOUNT,(MODE(N),N=1,NBUF4)
      STOP
 1030 IF(NCOUNT.GE.NTH)   GO TO 1040
      IWRK   = IANDNM+2
      IX     = IBUFFR(IWRK)
```

```
FILE = J3679.DPOOL2.FORT77                    DATE     84/03/15(THURSDAY)      TIME 15:00:36  PAGE 00

                IF(IX.GT.0)         GO TO 1000                                            00000350
                WRITE(6,9030)                                                             00000360
                STOP                                                                      00000370
           1040 WRITE(6,9040)                                                             00000380
                STOP                                                                      00000390
           9000 FORMAT('0*****   ERROR-STOP ( DATA POOL ALLOCATION ERROR ) *****'         00000400
               1       /'        LOGICAL UNIT NUMBER ----- ',I5                           00000410
               2       /'        NODE LELEL          ----- ',I5                           00000420
               3       /'        NODE NAME           ----- ',10(1X,A4))                   00000430
           9001 FORMAT('0+++++   ERROR MESSAGE   +++++')                                  00000440
           9010 FORMAT('0        THIS DIRECTORY ( ',A4,' ) DOSE NOT HAVE  SUBDIREC00000450
               1TORY')                                                                    00000460
           9020 FORMAT('0        ALL NODE NAMES IN LEVEL',I2,                             00000470
               1       /(10(4X,A4)))                                                      00000480
           9030 FORMAT('0        LOWER NODE NAME NOT EXISTS')                             00000490
           9040 FORMAT('0        THIS NODE NAME EXISTS IN DATA-POOL',                     00000500
               1    /'0 +++++  COMPUTER MULFUNCTION   +++++++')                           00000510
                END                                                                       00000520

**************
** PAGET    **
**************

                SUBROUTINE PAGET(NASELF)                                                  00000010
          C  SET CURRENT ADDRESS TO NASELF                                                00000020
                COMMON /DPCONT/ LCONTR,NCONTR,ICONTR(40,99),IX,NSUBDS,NDSTAT              00000030
                NASELF=IX                                                                 00000040
                RETURN                                                                    00000050
                END                                                                       00000060

**************
** PASTO    **
**************

                SUBROUTINE PASTO(NASELF)                                                  00000010
                COMMON /DPCONT/ LCONTR,NCONTR,ICONTR(40,99),IX,NSUBDS,NDSTAT              00000020
          C                                                                               00000030
                IX = NASELF                                                               00000040
                RETURN                                                                    00000050
                END                                                                       00000060

**************
** PDELT    **
**************

                SUBROUTINE PDELT(NUNIT,NODE,NTH,NRETUN)                                   00000100
          C                                                                               00000200
          C---- DECLARATION                                                               00000300
          C                                                                               00000400
                COMMON /DPCONT/ LCONTR,NCONTR,ICONTR(40,99),IX,NSUBDS,NDSTAT              00000500
                COMMON /DPWORK/ LBUFFR,LRECOD,IBUFFR(1000),NRECOD,NODE1,NODE2,            00000600
               *                NADWN,NADAT,NDASET,NDATE(2),NINFOM(5),NUTOLD,             00000700
               *                NTHOLD,NODOLD(10,2),NA1                                   00000800
                COMMON /DPDELT/ IBUFF2(1000)                                              00000900
                CHARACTER*4 CHR                                                           00001000

FILE = J3679.DPOOL2.FORT77                    DATE     84/03/15(THURSDAY)      TIME 15:00:36  PAGE 00
FILE = J3679.DPOOL2.FORT77                    DATE    84/03/15(THURSDAY)     TIME 15:00:36  PAGE 00

C
      DIMENSION NODE(1),NSDOLD(12)
C
C---- DELETE DIRECTORY AND DATA AND SUBDIRECTORY OF THE NODES WHICH IS
C     LOWER THAN NODE(NTH)
C
      IF(NTH.GT.0)                                GO TO 1010
          WRITE(6,7910)
          NRETUN=1
          GO TO 5900
 1010 IWFILE = NUNIT
      READ(NUNIT,REC=1) (ICONTR(LP1,IWFILE),LP1=1,LCONTR)
      LRECOD=ICONTR(26,IWFILE)
C  WRITE FLAG ON
      ICONTR(24,IWFILE)=1
      WRITE(NUNIT,REC=1) (ICONTR(LP1,IWFILE),LP1=1,LCONTR)
      NCOUNT = 0
      IX = ICONTR(21,IWFILE)
C
C---- SEARCH THE SUBDIRECTORY OF THE LAST NODE
C
 1100 NCOUNT = NCOUNT+1
      READ(NUNIT,REC=IX)          (IBUFFR(LP1),LP1=1,LRECOD)
      IF(IBUFFR(4).GT.0)                          GO TO 1110
          NRETUN = 1
          GO TO 5800
C
 1110 IANDNM = 5
      DO 1210  LP1=1,IBUFFR(4)
          IF(NODE(NCOUNT).EQ.IBUFFR(IANDNM))      GO TO 1300
          IANDNM = IANDNM+12
 1210 CONTINUE
      NRETUN = 1
                                                  GO TO 5800
C
 1300 IF(NCOUNT.GE.NTH)                           GO TO 2000
      IWRK=IANDNM+2
      IX = IBUFFR(IWRK)
                                                  GO TO 1100
C
C---- SAVE SUBDIRECTORY AND DELETE DIRECTORY AND DATA BY CALLING
C     SUBROUTINE SUBDLT
C
C2000 NASUBD = IX-1
 2000 NASUBD = IX
      NOSUBD = LP1
*     WRITE(6,7810)      NASUBD,NOSUBD
*7810 FORMAT(1H ,5X,'----- NASUBD =',I13,
*    */6X,'----- NOSUBD =',I13)
C**** STORE SUBDIRECTORY TO NSDOLD(12)
      DO 2100  LP1=1,12
          IWRK = IANDNM+(LP1-1)
          NSDOLD(LP1)=IBUFFR(IWRK)
 2100 CONTINUE
C**** DELETE DIRECTORY AND DATA OF THE NODE(NTH)


FILE = J3679.DPOOL2.FORT77                    DATE    84/03/15(THURSDAY)     TIME 15:00:36  PAGE 00
```

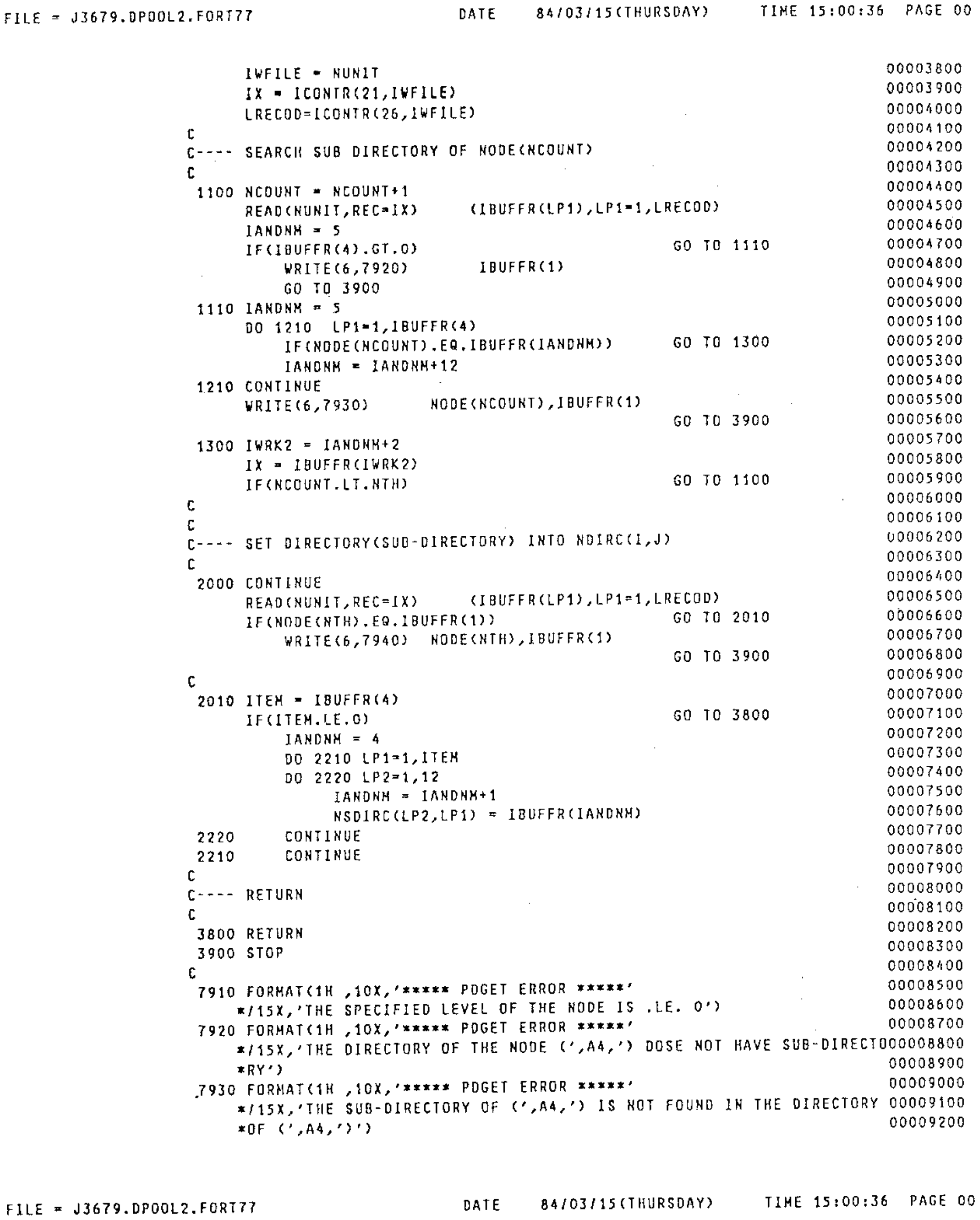


```
FILE = J3679.DPOOL2.FORT77                 DATE     84/03/15(THURSDAY)      TIME 15:00:36  PAGE 00

*     WRITE(6,7820)       NSDOLD
*7820 FORMAT(1H ,5X,'----- DIRECTORY(DEBUG) ',
*     */10X,'NODE1S =',A4
*     */10X,'NODE2S =',A4
*     */10X,'NADWN  = ',I13
*     */10X,'NADAT  = ',I13
*     */10X,'NDASET = ',I13
*     */10X,'DATE   = ',2A4
*     */10X,'INFOM  = ',5(I13,'*'))
      CALL SUBDLT(NUNIT,NSDOLD)
C**** DELETE DIRECTORY AND DATA OF THE NODE WHICH IS LOWER THAN NODE(NTH
C     )
      NCOUNT = 0
 3000 NCOUNT=NCOUNT+1
      IX=NASUBD+NCOUNT
      IF(IX.GE.ICONTR(22,IWFILE))                   GO TO 4000
      READ(NUNIT,REC=IX)      (IBUFF2(LP1),LP1=1,LRECOD)
C     IF(IBUFF2(1).NE.'////')                       GO TO 3000
      WRITE(CHR,'(A4)') IBUFF2(1)
      IF(CHR.NE.'////')                             GO TO 3000
      IF(IBUFF2(4).LE.0)                            GO TO 3000
      NOSUBD=IBUFF2(4)
      IBUFF2(4)=0
      WRITE(NUNIT,REC=IX)     (IBUFF2(LP1),LP1=1,LRECOD)
          IANDNM = 5
          DO 3010  LP1=1,NOSUBD
          DO 3020  LP2=1,12
               IWRK = IANDNM+(LP2-1)
               NSDOLD(LP2) = IBUFF2(IWRK)
 3020     CONTINUE
          CALL SUBDLT(NUNIT,NSDOLD)
          IANDNM = IANDNM+12
 3010     CONTINUE
                                                    GO TO 3000
C
C---- DELETE SUBDIRECTORY OF THE NODE(NTH)
C
 4000 CONTINUE
      IX = NASUBD
      READ(NUNIT,REC=IX)      (IBUFFR(LP3),LP3=1,LRECOD)
      IF(IBUFFR(4).LE.NOSUBD)                       GO TO 4100
          MAXLOP = IBUFFR(4)-NOSUBD
          MOVTO = 5+(NOSUBD-1)*12
          MOVFRM = 5+NOSUBD*12
          DO 4010   LP2=1,MAXLOP
          DO 4020   LP3=1,12
               ITO = MOVTO+LP3-1
               IFROM = MOVFRM+LP3-1
               IBUFFR(ITO) = IBUFFR(IFROM)
 4020     CONTINUE
          MOVTO = MOVTO+12
          MOVFRM = MOVFRM+12
 4010     CONTINUE
C
 4100 IBUFFR(4) = IBUFFR(4)-1

FILE = J3679.DPOOL2.FORT77                 DATE     84/03/15(THURSDAY)      TIME 15:00:36  PAGE 00
```

```
C     IX = IX-1                                                         00012100
      WRITE(NUNIT,REC=IX)    (IBUFFR(LP3),LP3=1,LRECOD)                 00012200
      NRETUN=0                                                          00012300
 5800 ICONTR(24,IWFILE)=0                                               00012400
      WRITE(NUNIT,REC=1)     (ICONTR(LP3,IWFILE),LP3=1,LCONTR)          00012500
C                                                                       00012600
C---- RETURN                                                            00012700
C                                                                       00012800
 5900 RETURN                                                            00012900
C                                                                       00013000
 7910 FORMAT(1H ,10X,'***** PDELT ERROR *****'                          00013100
     */15X,'THE SPECIFIED LEVEL OF THE NODE IS .LE. 0')                 00013200
      END                                                               00013300
```

```
**************
** PDGET    **
**************
```

```
      SUBROUTINE PDGET(NUNIT,NODE,NTH,ITEM,NSDIRC)                      00000100
C-----------------------------------------------------------------------I00000200
C     PDGET                                                             I00000300
C   1. FUNCTION                                                         I00000400
C     (1) READ THE DIRECTORY OF NODE(NTH)                               I00000500
C                                                                       I00000600
C   2. INPUT                                                            I00000700
C     (1) ARGUMENT                                                      I00000800
C       1-1 NUNIT   : INPUT FILE NO OF DIRECTORY                        I00000900
C       1-2 NODE(I) : NODE NAME                                         I00001000
C       1-3 NTH     : THE DEPTH OF NODE                                 I00001100
C                                                                       I00001200
C   3. OUTPUT                                                           I00001300
C     (1) ARGUMENT                                                      I00001400
C       1-1 ITEM    : THE NUMBER OF THE LOWER NODE OF THE DIRECTORY     I00001500
C       1-2 NSDIRC(I,J)  : THE ARRAY THAT SUBDIRECTORIES ARE TO BE      I00001600
C                          STORED                                       I00001700
C   4. RESTRICTION                                                      I00001800
C     (1) NSDIRC(1,J)                                                   I00001900
C         I = 12 , J,GE.ITEM                                            I00002000
C-----------------------------------------------------------------------I00002100
C                                                                       00002200
C---- DECLARATION                                                       00002300
C                                                                       00002400
      COMMON /DPCONT/ LCONTR,NCONTR,ICONTR(40,99),IX,NSUBDS,NDSTAT      00002500
      COMMON /DPWORK/ LBUFFR,LRECOD,IBUFFR(1000),NRECOD,NODE1,NODE2,    00002600
     *                NADWN,NADAT,NDASET,NDATE(2),NINFOH(5),NUTOLD,     00002700
     *                NTHOLD,NODOLD(10,2),NA1                           00002800
C                                                                       00002900
      DIMENSION NODE(1),NSDIRC(12,1)                                    00003000
C                                                                       00003100
C---- CHECK INPUT VARIABLES AND SET INITIAL VALUES                      00003200
C                                                                       00003300
      IF(NTH.GE.1)                                 GO TO 1010           00003400
          WRITE(6,7910)                                                 00003500
                                                   GO TO 3900           00003600
 1010 NCOUNT = 0                                                        00003700
```



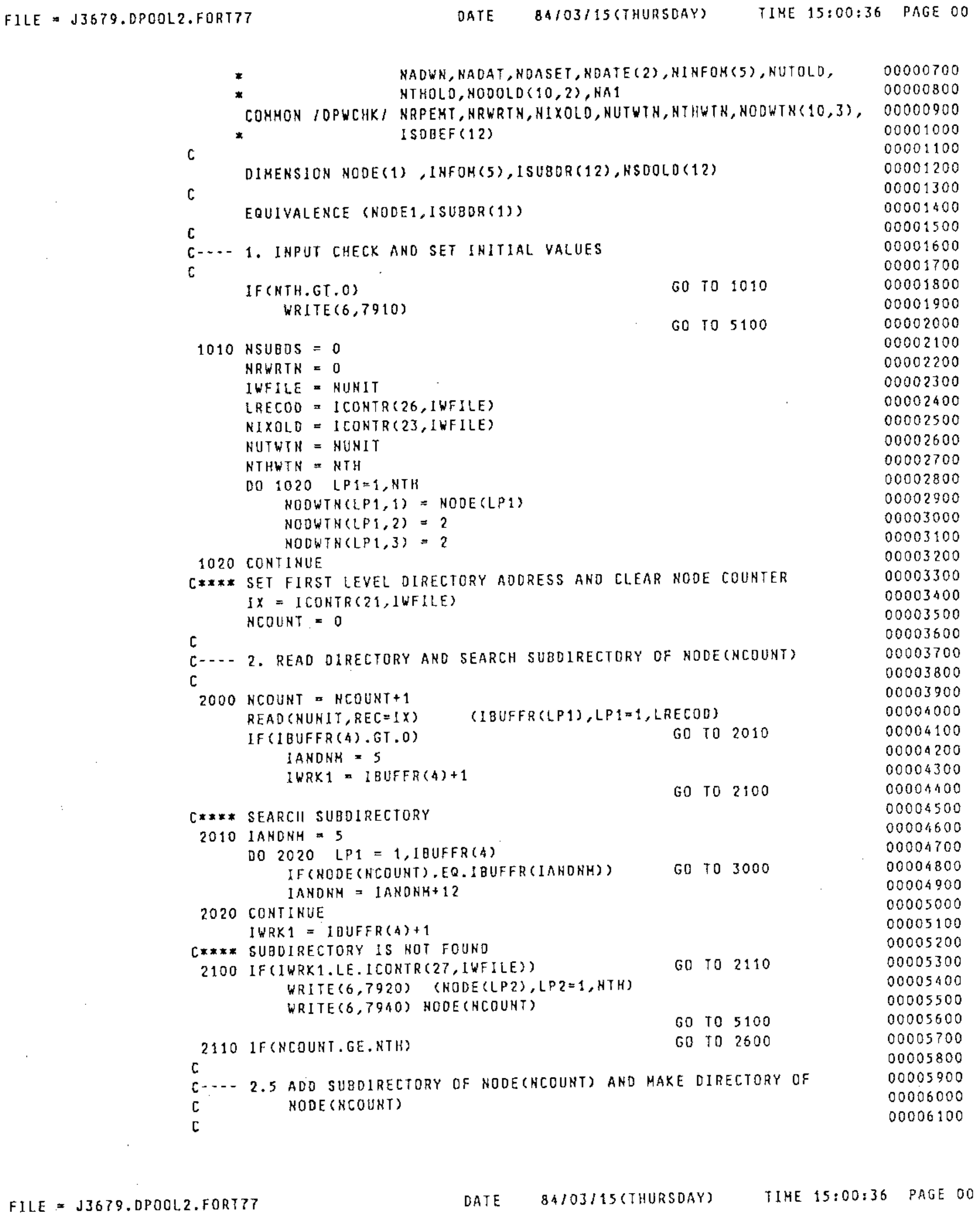
```
FILE = J3679.DPOOL2.FORT77          DATE    84/03/15(THURSDAY)     TIME 15:00:36  PAGE 00

      IWFILE = NUNIT
      IX = ICONTR(21,IWFILE)
      LRECOD=ICONTR(26,IWFILE)
C
C---- SEARCH SUB DIRECTORY OF NODE(NCOUNT)
C
 1100 NCOUNT = NCOUNT+1
      READ(NUNIT,REC=IX)     (IBUFFR(LP1),LP1=1,LRECOD)
      IANDNM = 5
      IF(IBUFFR(4).GT.0)                          GO TO 1110
          WRITE(6,7920)       IBUFFR(1)
          GO TO 3900
 1110 IANDNM = 5
      DO 1210  LP1=1,IBUFFR(4)
          IF(NODE(NCOUNT).EQ.IBUFFR(IANDNM))      GO TO 1300
          IANDNM = IANDNM+12
 1210 CONTINUE
      WRITE(6,7930)      NODE(NCOUNT),IBUFFR(1)
                                                  GO TO 3900
 1300 IWRK2 = IANDNM+2
      IX = IBUFFR(IWRK2)
      IF(NCOUNT.LT.NTH)                           GO TO 1100
C
C
C---- SET DIRECTORY(SUB-DIRECTORY) INTO NDIRC(I,J)
C
 2000 CONTINUE
      READ(NUNIT,REC=IX)     (IBUFFR(LP1),LP1=1,LRECOD)
      IF(NODE(NTH).EQ.IBUFFR(1))                  GO TO 2010
          WRITE(6,7940)  NODE(NTH),IBUFFR(1)
                                                  GO TO 3900
C
 2010 ITEM = IBUFFR(4)
      IF(ITEM.LE.0)                               GO TO 3800
          IANDNM = 4
          DO 2210 LP1=1,ITEM
          DO 2220 LP2=1,12
               IANDNM = IANDNM+1
               NSDIRC(LP2,LP1) = IBUFFR(IANDNM)
 2220     CONTINUE
 2210     CONTINUE
C
C---- RETURN
C
 3800 RETURN
 3900 STOP
C
 7910 FORMAT(1H ,10X,'***** PDGET ERROR *****'
     */15X,'THE SPECIFIED LEVEL OF THE NODE IS .LE. 0')
 7920 FORMAT(1H ,10X,'***** PDGET ERROR *****'
     */15X,'THE DIRECTORY OF THE NODE (',A4,') DOSE NOT HAVE SUB-DIRECTO
     *RY')
 7930 FORMAT(1H ,10X,'***** PDGET ERROR *****'
     */15X,'THE SUB-DIRECTORY OF (',A4,') IS NOT FOUND IN THE DIRECTORY
     *OF (',A4,')')

FILE = J3679.DPOOL2.FORT77          DATE    84/03/15(THURSDAY)     TIME 15:00:36  PAGE 00
```

FILE = J3679.DPOOL2.FORT77   DATE   84/03/15(THURSDAY)   TIME 15:00:36 PAGE 00

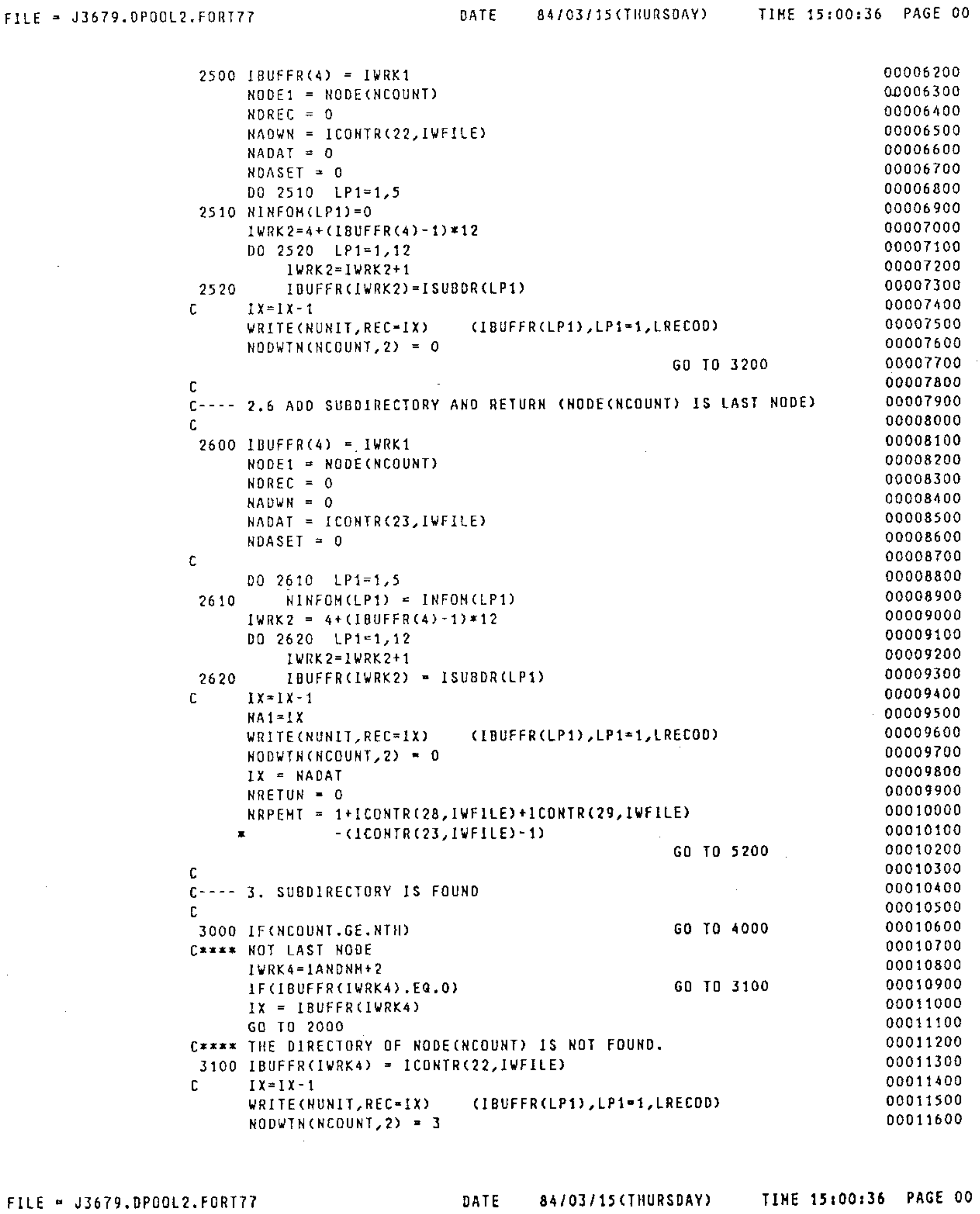
```
               7940 FORMAT(1H ,10X,'***** PDGET ERROR *****'
                   */15X,'THE DIRECTORY OF THE NODE (',A4,') CAN NOT BE FOUND',
                   */15X,'THE NODE NAME OF THE DIRECTORY IS (',A4,')')
              C
                    END

**************
** PFIND    **
**************

                    SUBROUTINE PFIND(NUNIT,NODE,NTH,NDIRC,LLL)
              C
              C---- DECLARATION
              C
                    COMMON /DPCONT/ LCONTR,NCONTR,ICONTR(40,99),IX,NSUBDS,NDSTAT
                    COMMON /DPWORK/ LBUFFR,LRECOD,IBUFFR(1000),NRECOD,NODE1,NODE2,
                   *                NADWN,NADAT,NDASET,NDATE(2),NINFOM(5),NUTOLD,
                   *                NTHOLD,NODOLD(10,2),NA1
              C
                    DIMENSION NODE(1),NDIRC(1)
              C
              C---- CHECK INPUT VALUE, SET INITIAL VALUE
              C
                    LLL=0
                    IF(NTH.GT.0)                                 GO TO 1010
                        WRITE(6,7910)
                        LLL=900
                        GO TO  4200
               1010 NCOUNT = 1
                    IWFILE = NUNIT
                    IX = ICONTR(21,IWFILE)
                    LRECOD=ICONTR(26,IWFILE)
                    IF(NUNIT.NE.NUTOLD)                          GO TO 2200
                    IF(NTHOLD.LE.1)                              GO TO 2200
              C
              C---- SEARCH SUBDIRECTORY ADRRESS BY OLD ACCESS TABLE NODOLD
              C
                    MAXLP1 =NTHOLD-1
                    DO 1020  LP1=1,MAXLP1
                        NCOUNT = NCOUNT+1
                        IF(NCOUNT.LE.NTH)                        GO TO 1030
                             NCOUNT=NCOUNT-1
                                                                 GO TO 2200
               1030     IF(NODOLD(NCOUNT,1).EQ.NODE(NCOUNT))     GO TO 1040
                                  NCOUNT=NCOUNT-1
                                                                 GO TO 2200
               1040     IX = NODOLD(NCOUNT,2)
               1020 CONTINUE
                                                                 GO TO 2200
              C
              C---- SEARCH SUBDIRECTORY ADDRESS OF NODE(NCOUNT) READING DIRECTORY OF
              C     NODE(NCOUNT-1)
              C
               2000 NCOUNT = NCOUNT+1
               2200 READ(NUNIT,REC=IX)     (IBUFFR(LP1),LP1=1,LRECOD)
```

FILE = J3679.DPOOL2.FORT77   DATE   84/03/15(THURSDAY)   TIME 15:00:36 PAGE 00

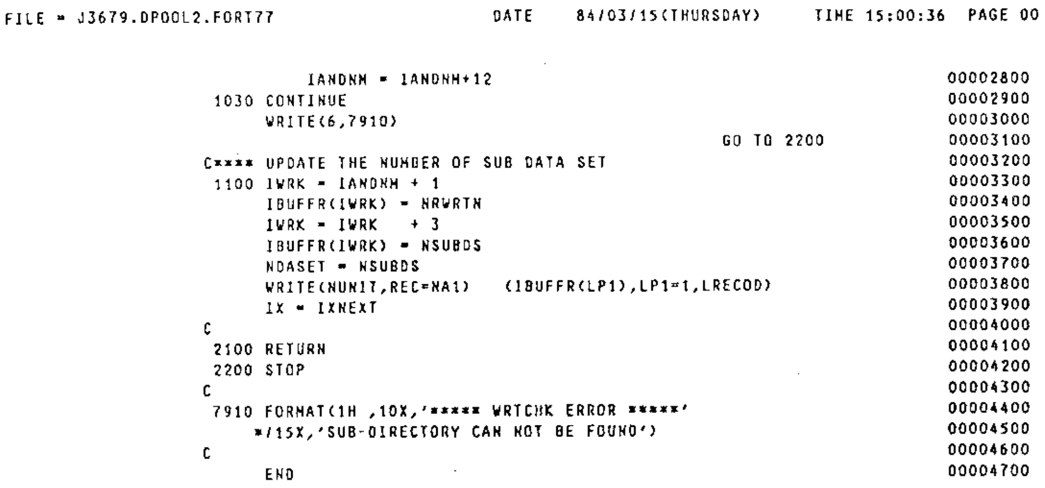
```
FILE = J3679.DPOOL2.FORT77                    DATE    84/03/15(THURSDAY)     TIME 15:00:36  PAGE 00

      IF(IBUFFR(4).GT.0)                             GO TO 2300
C         WRITE(6,7920)        IBUFFR(1)
      LLL=800
      GO TO  4100
 2300 IANDNH=5
      DO 2310 LP1=1,IBUFFR(4)
         IF(NODE(NCOUNT).EQ.IBUFFR(IANDNH))        GO TO 2400
         IANDNH=IANDNH+12
 2310 CONTINUE
C     WRITE(6,7940)
      LLL=100+NCOUNT
      GO TO  4100
 2400 NODOLD(NCOUNT,1)=IBUFFR(IANDNH)
C     NODOLD(NCOUNT,2) = IX-1
      NODOLD(NCOUNT,2) = IX
      IF(NCOUNT.GE.NTH)                              GO TO 3000
          IWRK = IANDNH+2
          IX = IBUFFR(IWRK)
          IF(IX.GT.0)                                GO TO 2500
C              WRITE(6,7930)        NODE(NCOUNT)
              LLL=200+NCOUNT
              GO TO  4100
 2500     CONTINUE
          GO TO 2000
C
C---- SET SUBDIRECTORY TO NDIRC AND SET DATA ADRRESS TO IX
C
 3000 DO 3010  LP1=1,12
          NDIRC(LP1) = IBUFFR(IANDNH)
          IANDNH = IANDNH+1
 3010 CONTINUE
      NA1= IX
      IX = NDIRC(4)
      NUTOLD = NUNIT
      NTHOLD = NTH
C
 4100 RETURN
 4200 STOP
C
 7910 FORMAT(1H ,10X,'***** PFIND ERROR *****'
     */15X,'THE SPECIFIED LEVEL OF THE NODE IS .LE. 0')
 7920 FORMAT(1H ,10X,'***** PFIND ERROR *****'
     */15X,'THE DIRECTORY DOES NOT HAVE SUB-DIRECTORY'
     */15X,' THE NODE NAME OF THE DIRECTORY IS (',A4,')')
 7930 FORMAT(1H ,10X,'***** PFIND ERROR *****'
     */15X,'THE ADRRESS OF THE DIRECTORY IS .LE. 0'
     */15X,'THE NODE NAME OF THE DIRECTORY IS (',A4,')')
 7940 FORMAT(1H ,10X,'***** PFIND ERROR *****',
     */15X,'NODE NAME INPUT ERROR')
      END

**************
** PINIT    **
**************

FILE = J3679.DPOOL2.FORT77                    DATE    84/03/15(THURSDAY)     TIME 15:00:36  PAGE 00
FILE = J3679.DPOOL2.FORT77                DATE    84/03/15(THURSDAY)     TIME 15:00:36  PAGE 00

      SUBROUTINE PINIT (NUNIT,NDIRCT,LENGTH,ITITLE)                     00000010
C                                                                       00000020
C-------- 0. DECLRATION                                                 00000030
C                                                                       00000040
      COMMON /DPCONT/ LCONTR,NCONTR,ICONTR(40,99),IX,NSUBDS,NDSTAT      00000050
C                                                                       00000060
      COMMON /DPWORK/ LBUFFR,LRECOD,IBUFFR(1000),NRECOD,NODE1,NODE2,    00000070
     *                NADWN,NADAT,NDASET,NDATE(2),NINFOM(5),NUTOLD,     00000080
     *                NTHOLD,NODOLD(10,2),NA1                           00000090
      COMMON /DPEMSG/ IFLAG,NERNO                                       00000100
C                                                                       00000110
      DIMENSION ITITLE(1)                                               00000120
      CHARACTER*4 BLANK                                                 00000121
      DATA BLANK/'    '/                                                00000122
      EXTERNAL SETMSG                                                   00000130
      CALL ERRSET(232,0,-1,1,SETMSG)                                    00000140
C                                                                       00000150
C-------- 1. DECLARATION OF DIRECT ACCESS FILE                          00000160
C                                                                       00000170
      LRECL=4*LENGTH                                                    00000180
      OPEN(NUNIT,ACCESS='DIRECT',RECL=LRECL)                            00000190
C                                                                       00000210
C-------- 2. SET CONTROL RECORD AND OUTPUT TO FILE NUNIT                00000220
C                                                                       00000230
      IWFILE = NUNIT                                                    00000240
      LCONTR=40                                                         00000250
      NCONTR=99                                                         00000260
      LBUFFR=1000                                                       00000270
      LRECOD=LENGTH                                                     00000280
      NRECOD=50000                                                      00000290
      IF(LRECOD.GT.LBUFFR)                          GO TO 3000          00000300
      NUTOLD=0                                                          00000310
      NTHOLD=0                                                          00000320
      DO 100 I=1,10                                                     00000330
      READ(BLANK,'(A4)') NODOLD(I,1)                                    00000340
      NODOLD(I,2)=0                                                     00000341
  100 CONTINUE                                                          00000342
C**** 2-1. SET TITLE                                                    00000350
      DO 2120 LP1 = 1,20                                                00000400
          ICONTR(LP1,IWFILE) = ITITLE(LP1)                              00000410
 2120 CONTINUE                                                          00000420
C**** 2-2. SET AND INITIALIZE OTHER CONTROL VARIABLES                   00000430
      ICONTR(21,IWFILE) = 2                                             00000440
      ICONTR(22,IWFILE) = 3                                             00000450
      ICONTR(23,IWFILE) = 2+NDIRCT                                      00000460
      IF(ICONTR(23,IWFILE).LE.NRECOD)                GO TO 2210         00000470
          WRITE(6,7900)       NUNIT,NRECOD                              00000480
          STOP                                                          00000490
 2210 CONTINUE                                                          00000500
      ICONTR(24,IWFILE) = 0                                             00000510
      ICONTR(25,IWFILE) = 0                                             00000520
      ICONTR(26,IWFILE) = LRECOD                                        00000530
      ICONTR(27,IWFILE) = (LRECOD-4)/12                                 00000540
      ICONTR(28,IWFILE) = NDIRCT                                        00000550
      ICONTR(29,IWFILE) = NRECOD-NDIRCT-1                               00000560

FILE = J3679.DPOOL2.FORT77                DATE    84/03/15(THURSDAY)     TIME 15:00:36  PAGE 00
FILE = J3679.DPOOL2.FORT77                  DATE    84/03/15(THURSDAY)     TIME 15:00:36  PAGE 00

      ICONTR(30,IWFILE) = 1
      ICONTR(31,IWFILE) = 0
      DO 2220 LP1 = 32,40
         ICONTR(LP1,IWFILE) = 0
 2220 CONTINUE
C**** 2-3. WRITE CONTROL SECTION
      WRITE(NUNIT,REC=1)      (ICONTR(LP1,IWFILE),LP1=1,LCONTR)
C  RESET NO. OF INITIALIZED RECORD
      IFLAG=0
      DO 300 N=2,NRECOD
      READ(NUNIT,REC=N)
      IF(IFLAG.EQ.1)                GO TO 350
  300 CONTINUE
      N=NRECOD+1
  350 NRECOD=N-1
      ICONTR(29,IWFILE)=NRECOD-NDIRCT-1
      WRITE(6,7000) NRECOD
      WRITE(6,7020)
      WRITE(6,7030) (ICONTR(LP1,IWFILE),LP1=1,18)
      WRITE(6,7040)  ICONTR(21,IWFILE)
      WRITE(6,7050)  ICONTR(22,IWFILE)
      WRITE(6,7060)  ICONTR(23,IWFILE)
      WRITE(6,7062)  ICONTR(24,IWFILE)
      WRITE(6,7064)  ICONTR(25,IWFILE)
      WRITE(6,7070)  ICONTR(26,IWFILE)
      WRITE(6,7080)  ICONTR(27,IWFILE)
      WRITE(6,7090)  ICONTR(28,IWFILE)
      WRITE(6,7100)  ICONTR(29,IWFILE)
      WRITE(6,7110)  ICONTR(30,IWFILE)
      WRITE(6,7120)  ICONTR(31,IWFILE)
      WRITE(NUNIT,REC=1) (ICONTR(LP1,IWFILE),LP1=1,LCONTR)
C
C-------- 3. MAKE FIRST LEVEL NODE DIRECTORY
C
      READ(BLANK,'(A4)') IBUFFR(1)
      IBUFFR(2) = 0
      IBUFFR(3) = 0
      IBUFFR(4) = 0
C
      WRITE(NUNIT,REC=2)  (IBUFFR(LP1),LP1=1,4)
C
C-------- 4. SET DATE
C
      CALL DATE(NDATE)
      RETURN
C
C-------- 5. ERROR STOP
C
 3000 WRITE(6,7800) LRECOD,LBUFFR
      STOP
C
C-------- 6. FORMAT
C
 7000 FORMAT('0*** MESSAGE FROM PINIT ***'/
     *       '      NO. OF INITIALIZED RECORD IS ',I5)

FILE = J3679.DPOOL2.FORT77                  DATE    84/03/15(THURSDAY)     TIME 15:00:36  PAGE 00
FILE = J3679.DPOOL2.FORT77                DATE    84/03/15(THURSDAY)     TIME 15:00:36  PAGE 00

 7020 FORMAT(1H0,'*********   C O N T R O L   S E C T I O N   *********'00001070
     *      /1H0,5X,'COL.')                                             00001080
 7030 FORMAT(1H0,4X,'1-18  TITLE    : '/1H0,6X,18A4)                    00001090
 7040 FORMAT(1H0,6X,'21  ADDRESS FOR THE DIRECTORY OF FIRST LEVEL NODE 00001100
     *: ',I10)                                                          00001110
 7050 FORMAT(1H0,6X,'22  HEAD ADDRESS FOR THE VACANT DIRECTORY AREA     00001120
     *: ',I10)                                                          00001130
 7060 FORMAT(1H0,6X,'23  HEAD ADDRESS FOR THE VACANT DATA AREA          00001140
     *: ',I10)                                                          00001150
 7062 FORMAT(1H0,6X,'24  WRITE FLAG                                     00001160
     *: ',I10)                                                          00001170
 7064 FORMAT(1H0,6X,'25  READ FLAG (NOT USED)                           00001180
     *: ',I10)                                                          00001190
 7070 FORMAT(1H0,6X,'26  LENGTH OF THE ONE PHYSICAL RECORD              00001200
     *: ',I10)                                                          00001210
 7080 FORMAT(1H0,6X,'27  MAXIMUM NUMBER OF THE SAME LEVEL NODE          00001220
     *: ',I10)                                                          00001230
 7090 FORMAT(1H0,6X,'28  SIZE OF THE DIRECTORY SECTION                  00001240
     *: ',I10)                                                          00001250
 7100 FORMAT(1H0,6X,'29  SIZE OF THE DATA SECTION                       00001260
     *: ',I10)                                                          00001270
 7110 FORMAT(1H0,6X,'30  REAL NUMBER OF THE DIRECTORY RECORDS           00001280
     *: ',I10)                                                          00001290
 7120 FORMAT(1H0,6X,'31  REAL NUMBER OF THE DATA SET RECORDS            00001300
     *: ',I10)                                                          00001310
 7800 FORMAT(1H ,10X,'***** PINIT ERROR *****'                          00001320
     */15X,'RECORD LENGTH IS TOO LONG. LENGTH=',I4,' MAX LENGTH=',I4)   00001330
 7900 FORMAT(1H ,10X,'***** PINIT ERROR *****'                          00001331
     */15X,'THE NUMBER OF DATA RECORD IS TOO LARGE       UNIT =',       00001332
     *I3/15X,'TOTAL RECORD NO =',I5)                                    00001340
C                                                                       00001350
      END                                                               00001360

**************
** POPEN    **
**************

      SUBROUTINE POPEN(NUNIT,JCONTR)                                    00000010
C                                                                       00000020
C-------- 0. DECLARATION                                                00000030
C                                                                       00000040
      COMMON /DPCONT/ LCONTR,NCONTR,ICONTR(40,99),IX,NSUBDS,NDSTAT      00000050
C                                                                       00000060
      COMMON /DPWORK/ LBUFFR,LRECOD,IBUFFR(1000),NRECOD,NODE1,NODE2,    00000070
     *                NADWN,NADAT,NDASET,NDATE(2),NINFOM(5),NUTOLD,     00000080
     *                NTHOLD,NODOLD(10,2),NA1                           00000090
      DIMENSION       JCONTR(1)                                         00000100
      CHARACTER DDNAME*8,BLANK*4                                        00000110
      DATA DDNAME/'FT  F001'/                                           00000120
      DATA BLANK/'    '/                                                00000121
C                                                                       00000130
C-------- 1. DECLARATION OF DIRECT ACCESS FILE                          00000140
C                                                                       00000150
C   GET RECORD LENGTH OF THIS DATA POOL BEFORE OPEN                     00000160
      N1=NUNIT/10                                                       00000170

FILE = J3679.DPOOL2.FORT77                DATE    84/03/15(THURSDAY)     TIME 15:00:36  PAGE 00
FILE = J3679.DPOOL2.FORT77                  DATE    84/03/15(THURSDAY)     TIME 15:00:36  PAGE 00

      WRITE(DDNAME(3:3),'(I1)') N1
      N2=NUNIT-N1*10
      WRITE(DDNAME(4:4),'(I1)') N2
      CALL GETDCB(DDNAME,LRECL,LBLKS,RECFM,DSORG,IRCD)
      IF(IRCD.NE.0)                      GO TO 1000
C  OPEN DATA POOL
      OPEN(NUNIT,ACCESS='DIRECT',RECL=LBLKS)
C
C-------- 2. READ CONTROL SECTION AND SET DATE
C
      LCONTR=40
      IWFILE=NUNIT
      READ(NUNIT,REC=1) (ICONTR(I,IWFILE),I=1,LCONTR)
      NCONTR=99
      LBUFFR=1000
      LRECOD=ICONTR(26,IWFILE)
      NRECOD=1+ICONTR(28,IWFILE)+ICONTR(29,IWFILE)
      NUTOLD=0
      NTHOLD=0
      DO 200 I=1,10
      READ(BLANK,'(A4)') NODOLD(I,1)
  200 NODOLD(I,2)=0
      DO 300 L=1,LCONTR
      JCONTR(L)=ICONTR(L,IWFILE)
  300 CONTINUE
      CALL DATE(NDATE)
C
C-------- 3. RETURN
C
      RETURN
 1000 WRITE(6,7000) DDNAME
      STOP
C
C-------- 4. FORMAT
C
 7000 FORMAT('0 **** POPEN ERROR   DDNAME=',A8,' IS NOT ALLOCATED')
      END

**************
** PREAD    **
**************

      SUBROUTINE PREAD(NUNIT,NAME1,NAME2,ICH,NASBD,NOSBDS,NOARY,NDATA)
C
C---- DECLARATION
C
      COMMON /DPCONT/ LCONTR,NCONTR,ICONTR(40,99),IX,NSUBDS,NDSTAT
      DIMENSION ICH(1),NDATA(1)
C
C---- READ CONTROL SECTION OF SUB-DATA SET
C
      IXSTAT = IX
      READ(NUNIT,REC=IX) NAME1,NAME2,(ICH(LP1),LP1=1,20),NASBD,NOSBDS,
     *                   NOARY,(NDATA(LP1),LP1=1,NOARY)
C

FILE = J3679.DPOOL2.FORT77                  DATE    84/03/15(THURSDAY)     TIME 15:00:36  PAGE 00
FILE = J3679.DPOOL2.FORT77                DATE    84/03/15(THURSDAY)     TIME 15:00:36  PAGE 00

              IX = IXSTAT
              RETURN
              END

 **************
 ** PREAD1   **
 **************

              SUBROUTINE PREAD1(NUNIT,ICH,NDATA1,DATA1)
        C------------------------------------------------------------------------
        C     1. FUNCTION
        C      (1) READ SUB DATA SET WHICH CONSISTS OF ONE ARRAY
        C
        C     2. INPUT
        C      (1) NUNIT    : FILE UNIT NO.                             (A)
        C      (2) IX       : READ-STARTING ADRRESS                     (C/DPCONT/)
        C      (3) ICH(20)  : THE TITTLE OF DATA SET                    (F NUNIT)
        C      (4) NDATA1   : THE SIZE OF DATA ARRAY                    (F NUNIT)
        C      (5) DATA1    : THE DATA ARRAY                            (F NUNIT)
        C
        C     3. OUTPUT
        C      (1) ICH(20)
        C      (2) NDATA1
        C      (3) DATA1
        C      (4) IX       : READ-STARTING POINT OF THE NEXT SUB
        C                     DATA SET
        C------------------------------------------------------------------------
        C
        C---- DECLARATION
        C
              COMMON /DPCONT/ LCONTR,NCONTR,ICONTR(40,99),IX,NSUBDS,NDSTAT
              COMMON /DPWORK/ LBUFFR,LRECOD,IBUFFR(1000),NRECOD,NODE1,NODE2,
             *                NADWN,NADAT,NDASET,NDATE(2),NINFOM(5),NUTOLD,
             *                NTHOLD,NODOLD(10,2),NA1
        C
              DIMENSION ICH(1),DATA1(1),NDATA(4)
        C
              IXOLD=IX
              READ(NUNIT,REC=IX) NODWK1,NODWK2,(ICH(LP1),LP1=1,20),NA1WK,NSDSWK,
             *               NOAWK,(NDATA(N),N=1,NOAWK),
             *               (DATA1(LP1),LP1=1,NDATA(1))
              NDATA1=NDATA(1)
        C     IF(NOAWK.EQ.1)                  GO TO 1000
        C  SET IX TO HEAD POSITION OF THE NEXT SUB DATA SET
              NWORD=25+NOAWK
              DO 100 N=1,NOAWK
          100 NWORD=NWORD+NDATA(N)
              NRECD=NWORD/LRECOD
              IF(MOD(NWORD,LRECOD).EQ.0)      GO TO 200
              NRECD=NRECD+1
          200 IX=IXOLD+NRECD
         1000 RETURN
              END

FILE = J3679.DPOOL2.FORT77                DATE    84/03/15(THURSDAY)     TIME 15:00:36  PAGE 00
FILE = J3679.DPOOL2.FORT77                    DATE    84/03/15(THURSDAY)      TIME 15:00:36  PAGE 00


     **************
     ** PREAD2   **
     **************

            SUBROUTINE PREAD2(NUNIT,ICM,NDATA1,DATA1,NDATA2,DATA2)
      C
      C---- DECLARATION
      C
            COMMON /DPCONT/ LCONTR,NCONTR,ICONTR(40,99),IX,NSUBDS,NDSTAT
            COMMON /DPWORK/ LBUFFR,LRECOD,IBUFFR(1000),NRECOD,NODE1,NODE2,
           *                NADWN,NADAT,NDASET,NDATE(2),NINFOM(5),NUTOLD,
           *                NTHOLD,NODOLD(10,2),NA1
      C
            DIMENSION ICM(1),DATA1(1),DATA2(1),NDATA(4)
      C
            IXOLD=IX
            READ(NUNIT,REC=IX) NODWK1,NODWK2,(ICM(LP1),LP1=1,20),NA1WK,NSDSWK,
           *               NOAWK,(NDATA(N),N=1,NOAWK),
           *               (DATA1(LP1),LP1=1,NDATA(1)),
           *               (DATA2(LP1),LP1=1,NDATA(2))
            NDATA1=NDATA(1)
            NDATA2=NDATA(2)
      C     IF(NOAWK.EQ.2)                   GO TO 1000
      C  SET IX TO HEAD POSITION OF THE NEXT SUB DATA SET
            NWORD=25+NOAWK
            DO 100 N=1,NOAWK
        100 NWORD=NWORD+NDATA(N)
            NRECD=NWORD/LRECOD
            IF(MOD(NWORD,LRECOD).EQ.0)       GO TO 200
            NRECD=NRECD+1
        200 IX=IXOLD+NRECD
       1000 RETURN
            END

     **************
     ** PREAD3   **
     **************

            SUBROUTINE PREAD3(NUNIT,ICM,NDATA1,DATA1,NDATA2,DATA2,NDATA3,
           *                  DATA3)
      C
      C---- DECLARATION
      C
            COMMON /DPCONT/ LCONTR,NCONTR,ICONTR(40,99),IX,NSUBDS,NDSTAT
            COMMON /DPWORK/ LBUFFR,LRECOD,IBUFFR(1000),NRECOD,NODE1,NODE2,
           *                NADWN,NADAT,NDASET,NDATE(2),NINFOM(5),NUTOLD,
           *                NTHOLD,NODOLD(10,2),NA1
      C
            DIMENSION ICM(1),DATA1(1),DATA2(1),DATA3(1),NDATA(4)
      C
            IXOLD=IX
            READ(NUNIT,REC=IX) NODWK1,NODWK2,(ICM(LP1),LP1=1,20),NA1WK,NSDSWK,
           *               NOAWK,(NDATA(N),N=1,NOAWK),
           *               (DATA1(LP1),LP1=1,NDATA(1)),


FILE = J3679.DPOOL2.FORT77                    DATE    84/03/15(THURSDAY)      TIME 15:00:36  PAGE 00
FILE = J3679.DPOOL2.FORT77                 DATE     84/03/15(THURSDAY)      TIME 15:00:36  PAGE 00

     *           (DATA2(LP1),LP1=1,NDATA(2)),
     *           (DATA3(LP1),LP1=1,NDATA(3))
      NDATA1=NDATA(1)
      NDATA2=NDATA(2)
      NDATA3=NDATA(3)
C     IF(NOAWK.EQ.3)                    GO TO 1000
C  SET IX TO HEAD POSITION OF THE NEXT SUB DATA SET
      NWORD=25+NOAWK
      DO 100 N=1,NOAWK
  100 NWORD=NWORD+NDATA(N)
      NRECD=NWORD/LRECOD
      IF(MOD(NWORD,LRECOD).EQ.0)        GO TO 200
      NRECD=NRECD+1
  200 IX=IXOLD+NRECD
 1000 RETURN
      END

**************
** PREAD4   **
**************

      SUBROUTINE PREAD4(NUNIT,ICH,NDATA1,DATA1,NDATA2,DATA2,
     *                  NDATA3,DATA3,NDATA4,DATA4)
C
C---- DECLARATION
C
      COMMON /DPCONT/ LCONTR,NCONTR,ICONTR(40,99),IX,NSUBDS,NDSTAT
      COMMON /DPWORK/ LBUFFR,LRECOD,IBUFFR(1000),NRECOD,NODE1,NODE2,
     *                NADWN,NADAT,NDASET,NDATE(2),NINFOM(5),NUTOLD,
     *                NTHOLD,NODOLD(10,2),NA1
C
      DIMENSION ICH(1),DATA1(1),DATA2(1),DATA3(1),DATA4(1),NDATA(4)
C
      IXOLD=IX
      READ(NUNIT,REC=IX) NODWK1,NODWK2,(ICH(LP1),LP1=1,20),NA1WK,NSDSWK,
     *           NOAWK,(NDATA(N),N=1,NOAWK),
     *           (DATA1(LP1),LP1=1,NDATA(1)),
     *           (DATA2(LP1),LP1=1,NDATA(2)),
     *           (DATA3(LP1),LP1=1,NDATA(3)),
     *           (DATA4(LP1),LP1=1,NDATA(4))
      NDATA1=NDATA(1)
      NDATA2=NDATA(2)
      NDATA3=NDATA(3)
      NDATA4=NDATA(4)
      NWORD=25+NOAWK
      DO 100 N=1,NOAWK
  100 NWORD=NWORD+NDATA(N)
      NRECD=NWORD/LRECOD
      IF(MOD(NWORD,LRECOD).EQ.0)        GO TO 200
      NRECD=NRECD+1
  200 IX=IXOLD+NRECD
 1000 RETURN
      END
FILE = J3679.DPOOL2.FORT77                 DATE     84/03/15(THURSDAY)      TIME 15:00:36  PAGE 00
FILE = J3679.DPOOL2.FORT77                    DATE     84/03/15(THURSDAY)      TIME 15:00:36  PAGE 00


     **************
     ** PRITE    **
     **************

      SUBROUTINE PRITE(NUNIT,ICM)
C
C---- DECLARATION
C
      COMMON /DPCONT/ LCONTR,NCONTR,ICONTR(40,99),IX,NSUBDS,NOSTAT
      COMMON /DPWORK/ LBUFFR,LRECOD,IBUFFR(1000),NRECOD,NODE1,NODE2,
     *                NADWN,NADAT,NDASET,NDATE(2),NINFOM(5),NUTOLD,
     *                NTHOLD,NODOLD(10,2),NA1
      COMMON /DPWCHK/ NRPEMT,NRWRTN,NIXOLD,NUTWTN,NTHWTN,NODWTN(10,3),
     *                ISDBEF(12)
C
      DIMENSION ICM(1)
C
C---- CHECK SIZE
C
      IXOLD=IX
      IWFILE = NUNIT
C
      NWORD=25
      NRECD = NWORD/LRECOD
      IF(MOD(NWORD,LRECOD).EQ.0)                    GO TO 1020
          NRECD = NRECD+1
 1020 ICHKNO = NRPEMT-(NRWRTN+NRECD)
      IF(ICHKNO.GE.0)                               GO TO 2000
          CALL DCLEAR
          WRITE(6,7920)       (NODWTN(LP9,1),LP9=1,NTHWTN)
          GO TO 5200
C
C---- WRITE DATA SET
C
 2000 NSUBDS = NSUBDS+1
      NOA = 0
      WRITE(NUNIT,REC=IX) NODE1,NODE2,(ICM(LP1),LP1=1,20),NA1,NSUBDS,NOA
      IX = IX + NRECD
C
C---- UPDATE CONTROL SECTION AND VARIABLES
C
      CALL WRTCHK(NUNIT,IXOLD)
C
C---- RETURN
C
 5100 RETURN
 5200 CONTINUE
      ICONTR(24,IWFILE)=0
      WRITE(NUNIT,REC=1) (ICONTR(LP1,IWFILE),LP1=1,LCONTR)
      STOP
C
 7920 FORMAT(1H ,10X,'***** PRITE ERROR *****'
     */15X,'DATA CAN NOT BE WRITTEN FOR THE LACK OF DOMAIN',
     */15X,10(A4,4X))

FILE = J3679.DPOOL2.FORT77                    DATE     84/03/15(THURSDAY)      TIME 15:00:36  PAGE 00
FILE = J3679.DPOOL2.FORT77                    DATE    84/03/15(THURSDAY)     TIME 15:00:36  PAGE 00

                  END

      **************
      ** PRITE1   **
      **************

                  SUBROUTINE PRITE1(NUNIT,ICH,NDATA1,DATA1)
            C
            C---- DECLARATION
            C
                  COMMON /DPCONT/ LCONTR,NCONTR,ICONTR(40,99),IX,NSUBDS,NDSTAT
                  COMMON /DPWORK/ LBUFFR,LRECOD,IBUFFR(1000),NRECOD,NODE1,NODE2,
                 *                NADWN,NADAT,NDASET,NDATE(2),NINFDM(5),NUTOLD,
                 *                NTHOLD,NODOLD(10,2),NA1
                  COMMON /DPWCHK/ NRPEMT,NRWRTN,NIXOLD,NUTWTN,NTHWTN,NODWTN(10,3),
                 *                ISDBEF(12)
            C
                  DIMENSION ICH(1),DATA1(1)
            C
            C---- CHECK SIZE .
            C
                  IXOLD = IX
                  IWFILE = NUNIT
                  IF(NDATA1.GT.0)                           GO TO 1010
                      WRITE(6,7910)       (NODWTN(LP9,1),LP9=1,NTHWTN)
                      CALL DCLEAR
                      GO TO 5200
             1010 NWORD = 26+NDATA1
                  NRECD = NWORD/LRECOD
                  IF(MOD(NWORD,LRECOD).EQ.0)                GO TO 1020
                      NRECD=NRECD+1
             1020 ICHKNO = NRPEMT-(NRWRTN+NRECD)
                  IF(ICHKNO.GE.0)                           GO TO 2000
                      CALL DCLEAR
                      WRITE(6,7920)       (NODWTN(LP9,1),LP9=1,NTHWTN)
                      GO TO 5200
            C
            C---- WRITE DATA SET
            C
             2000 NSUBDS=NSUBDS+1
                  NOA = 1
                  WRITE(NUNIT,REC=IX) NODE1,NODE2,(ICH(LP1),LP1=1,20),NA1,NSUBDS,
                 *            NOA,NDATA1,(DATA1(LP1),LP1=1,NDATA1)
                  IX = IX + NRECD
            C
            C---- UPDATE CONTROL SECTION AND VARIABLES
            C
                  CALL WRTCHK(NUNIT,IXOLD)
            C
            C---- RETURN
            C
             5100 RETURN
             5200 CONTINUE
                  ICONTR(24,IWFILE)=0
                  WRITE(NUNIT,REC=1) (ICONTR(LP1,IWFILE),LP1=1,LCONTR)

FILE = J3679.DPOOL2.FORT77                    DATE    84/03/15(THURSDAY)     TIME 15:00:36  PAGE 00
FILE = J3679.DPOOL2.FORT77                 DATE    84/03/15(THURSDAY)     TIME 15:00:36  PAGE 00

      STOP
C
 7910 FORMAT(1H ,10X,'***** PRITE1 ERROR *****'
     */15X,'THE SIZE OF DATA IS LESS OR EQUAL 0'
     */15X,10(A4,4X))
 7920 FORMAT(1H ,10X,'***** PRITE1 ERROR *****'
     */15X,'DATA CAN NOT BE WRITTEN FOR THE LACK OF DOMAIN'
     */15X,10(A4,4X))
      END

**************
** PRITE2   **
**************

      SUBROUTINE PRITE2(NUNIT,ICH,NDATA1,DATA1,NDATA2,DATA2)
C
C---- DECLARATION
C
      COMMON /DPCONT/ LCONTR,NCONTR,ICONTR(40,99),IX,NSUBDS,NDSTAT
      COMMON /DPWORK/ LBUFFR,LRECOD,IBUFFR(1000),NRECOD,NODE1,NODE2,
     *                NADWN,NADAT,NDASET,NDATE(2),NINFOM(5),NUTOLD,
     *                NTHOLD,NODOLD(10,2),NA1
      COMMON /DPWCHK/ NRPEMT,NRWRTN,NIXOLD,NUTWTN,NTHWTN,NODWTN(10,3),
     *                ISDBEF(12)
C
      DIMENSION ICH(1),DATA1(1),DATA2(1)
C
C---- CHECK SIZE
C
      IXOLD = IX
      IWFILE = NUNIT
C
      IF((NDATA1.GT.0).AND.(NDATA2.GT.0))         GO TO 1010
          WRITE(6,7910)       (NODWTN(LP9,1),LP9=1,NTHWTN)
          CALL DCLEAR
          GO TO 5200
 1010 NWORD=27+NDATA1+NDATA2
      NRECD=NWORD/LRECOD
      IF(MOD(NWORD,LRECOD).EQ.0)                  GO TO 1020
          NRECD=NRECD+1
 1020 ICHKNO = NRPEMT-(NRWRTN+NRECD)
      IF(ICHKNO.GE.0)                             GO TO 2000
          CALL DCLEAR
          WRITE(6,7920)       (NODWTN(LP9,1),LP9=1,NTHWTN)
          GO TO 5200
C
C---- WRITE DATA SET
C
 2000 NSUBDS=NSUBDS+1
      NOA=2
      WRITE(NUNIT,REC=IX) NODE1,NODE2,(ICH(LP1),LP1=1,20),NA1,NSUBDS,
     *            NOA,NDATA1,NDATA2,(DATA1(LP1),LP1=1,NDATA1),
     *            (DATA2(LP1),LP1=1,NDATA2)
      IX = IX + NRECD
C

FILE = J3679.DPOOL2.FORT77                 DATE    84/03/15(THURSDAY)     TIME 15:00:36  PAGE 00
FILE = J3679.DPOOL2.FORT77                DATE    84/03/15(THURSDAY)     TIME 15:00:36  PAGE 00

          C---- UPDATE CONTROL SECTION AND VARIABLES
          C
                CALL WRTCHK(NUNIT,IXOLD)
          C
          C---- RETURN
          C
           5100 RETURN
           5200 CONTINUE
                ICONTR(24,IWFILE)=0
                WRITE(NUNIT,REC=1) (ICONTR(LP1,IWFILE),LP1=1,LCONTR)
                STOP
          C
           7910 FORMAT(1H ,10X,'***** PRITE2 ERROR *****'
               */15X,'THE SIZE OF DATA IS LESS OR EQUAL 0'
               */15X,10(A4,4X))
           7920 FORMAT(1H ,10X,'***** PRITE2 ERROR *****' /
               *15X,'DATA CAN NOT BE WRITTEN FOR THE LACK OF DOMAIN'
               */15X,10(A4,4X))
                END

    **************
    ** PRITE3   **
    **************

                SUBROUTINE PRITE3(NUNIT,ICH,NDATA1,DATA1,NDATA2,DATA2,NDATA3,
               1                 DATA3)
          C---- DECLARATION
          C
                COMMON /DPCONT/ LCONTR,NCONTR,ICONTR(40,99),IX,NSUBDS,NDSTAT
                COMMON /DPWORK/ LBUFFR,LRECOD,IBUFFR(1000),NRECOD,NODE1,NODE2,
               *                NADWN,NADAT,NDASET,NDATE(2),NINFOM(5),NUTOLD,
               *                NTHOLD,NODOLD(10,2),NA1
                COMMON /DPWCHK/ NRPEMT,NRWRTN,NIXOLD,NUTWTN,NTHWTN,NODWTN(10,3),
               *                ISOBEF(12)
          C
                DIMENSION ICH(1),DATA1(1),DATA2(1),DATA3(1)
          C
          C---- CHECK SIZE
          C
                IXOLD=IX
                IWFILE = NUNIT
          C
                IF((NDATA1.GT.0).AND.(NDATA2.GT.0).AND.(NDATA3.GT.0))
               *                                        GO TO 1010
                    WRITE(6,7910)       (NODWTN(LP9,1),LP9=1,NTHWTN)
                    CALL DCLEAR
                    GO TO 5200
           1010 NWORD = 28+NDATA1+NDATA2+NDATA3
                NRECD = NWORD/LRECOD
                IF(MOD(NWORD,LRECOD).EQ.0)              GO TO 1020
                    NRECD = NRECD+1
           1020 ICHKNO = NRPEMT-(NRWRTN+NRECD)
                IF(ICHKNO.GE.0)                         GO TO 2000
                    CALL DCLEAR
                    WRITE(6,7920)       (NODWTN(LP9,1),LP9=1,NTHWTN)

FILE = J3679.DPOOL2.FORT77                DATE    84/03/15(THURSDAY)     TIME 15:00:36  PAGE 00
FILE = J3679.DPOOL2.FORT77                    DATE    84/03/15(THURSDAY)    TIME 15:00:36  PAGE 00

            GO TO 5200
C
C---- WRITE DATA SET
C
 2000 NSUBDS = NSUBDS+1
      NOA = 3
      WRITE(NUNIT,REC=IX) NODE1,NODE2,(ICH(LP1),LP1=1,20),NA1,NSUBDS,
     *             NOA,NDATA1,NDATA2,NDATA3,(DATA1(LP1),LP1=1,NDATA1),
     *             (DATA2(LP1),LP1=1,NDATA2),(DATA3(LP1),LP1=1,NDATA3)
      IX = IX + NRECO
C
C---- UPDATE CONTROL SECTION AND VARIABLES
C
      CALL WRTCHK(NUNIT,IXOLD)
C
C---- RETURN
C
 5100 RETURN
 5200 CONTINUE
      ICONTR(24,IWFILE)=0
      WRITE(NUNIT,REC=1)  (ICONTR(LP1,IWFILE),LP1=1,LCONTR)
      STOP
C
 7910 FORMAT(1H ,10X,'***** PRITE3 ERROR *****'
     */15X,'THE SIZE OF DATA IS LESS OR EQUAL 0'
     */15X,10(A4,4X))
 7920 FORMAT(1H ,10X,'***** PRITE3 ERROR *****'
     */15X,'DATA CAN NOT BE WRITTEN FOR THE LACK OF DOMAIN'
     */15X,10(A4,4X))
      END

***************
** PRITE4    **
***************

      SUBROUTINE PRITE4(NUNIT,ICH,NDATA1,DATA1,NDATA2,DATA2,NDATA3,
     *                  DATA3,NDATA4,DATA4)
C
C---- DECLARATION
C
      COMMON /DPCONT/ LCONTR,NCONTR,ICONTR(40,99),IX,NSUBDS,NDSTAT
      COMMON /DPWORK/ LBUFFR,LRECOD,IBUFFR(1000),NRECOD,NODE1,NODE2,
     *                NADWN,NADAT,NDASET,NDATE(2),NINFOH(5),NUTOLD,
     *                NTHOLD,NODOLD(10,2),NA1
      COMMON /DPWCHK/ NRPEMT,NRWRTN,NIXOLD,NUTWTN,NTHWTN,NODWTN(10,3),
     *                ISDBEF(12)
C
      DIMENSION ICH(1),DATA1(1),DATA2(1),DATA3(1),DATA4(1)
C
C---- CHECK SIZE
C
      IXOLD = IX
      IWFILE = NUNIT
C
      IF((NDATA1.GT.0).AND.(NDATA2.GT.0).AND.(NDATA3.GT.0).AND.

FILE = J3679.DPOOL2.FORT77                    DATE    84/03/15(THURSDAY)    TIME 15:00:36  PAGE 00
FILE = J3679.DPOOL2.FORT77                    DATE    84/03/15(THURSDAY)     TIME 15:00:36  PAGE 00

     *(NDATA4.GT.0))                                 GO TO 1010
          WRITE(6,7910)       (NODWTN(LP9,1),LP9=1,NTHWTN)
          CALL DCLEAR
          GO TO 5200
 1010 NWORD = 29+NDATA1+NDATA2+NDATA3+NDATA4
      NRECD = NWORD/LRECOD
      IF(MOD(NWORD,LRECOD).EQ.0)                     GO TO 1020
          NRECD = NRECD+1
 1020 ICHKNO = NRPENT-(NRWRTN+NRECD)
      IF(ICHKNO.GE.0)                                GO TO 2000
          CALL DCLEAR
          WRITE(6,7920)
          GO TO 5200
C
C---- WRITE DATA SET
C
 2000 NSUBDS = NSUBDS+1
      NOA = 4
      WRITE(NUNIT,REC=IX) NODE1,NODE2,(ICM(LP1),LP1=1,20),NA1,NSUBDS,
     *            NOA,NDATA1,NDATA2,NDATA3,NDATA4,(DATA1(LP1),LP1=1,
     *            NDATA1),(DATA2(LP1),LP1=1,NDATA2),
     *            (DATA3(LP1),LP1=1,NDATA3),
     *            (DATA4(LP1),LP1=1,NDATA4)
      IX = IX + NRECD
C
C---- UPDATE CONTROL SECTION AND VARIABLES
C
      CALL WRTCHK(NUNIT,IXOLD)
C
C---- RETURN
C
 5100 RETURN
 5200 CONTINUE
      ICONTR(24,IWFILE)=0
      WRITE(NUNIT,REC=1) (ICONTR(LP1,IWFILE),LP1=1,LCONTR)
      STOP
C
 7910 FORMAT(1H ,10X,'***** PRITE4 ERROR *****'
     */15X,'THE SIZE OF DATA IS LESS OR EQUAL 0'
     */15X,10(A4,4X))
 7920 FORMAT(1H ,10X,'***** PRITE4 ERROR *****'
     */15X,'THE SIZE OF DATA IS LESS OR EQUAL 0'
     */15X,10(A4,4X))
      END

**************
** PSET     **
**************

      SUBROUTINE PSET(NUNIT,NODE,NTH,INFOM,NUPDAT,NRETUN)
C
C----    DECLARATION
C
      COMMON /DPCONT/ LCONTR,NCONTR,ICONTR(40,99),IX,NSUBDS,NDSTAT
      COMMON /DPWORK/ LBUFFR,LRECOD,IBUFFR(1000),NRECOD,NODE1,NDREC,

FILE = J3679.DPOOL2.FORT77                    DATE    84/03/15(THURSDAY)     TIME 15:00:36  PAGE 00
```



```
FILE = J3679.DPOOL2.FORT77                DATE    84/03/15(THURSDAY)     TIME 15:00:36  PAGE 00

     *                NADWN,NADAT,NDASET,NDATE(2),NINFOM(5),NUTOLD,
     *                NTHOLD,NODOLD(10,2),NA1
      COMMON /DPWCHK/ NRPEMT,NRWRTN,NIXOLD,NUTWTN,NTHWTN,NODWTN(10,3),
     *                ISOBEF(12)
C
      DIMENSION NODE(1) ,INFOM(5),ISUBDR(12),NSDOLD(12)
C
      EQUIVALENCE (NODE1,ISUBDR(1))
C
C---- 1. INPUT CHECK AND SET INITIAL VALUES
C
      IF(NTH.GT.0)                             GO TO 1010
          WRITE(6,7910)
                                               GO TO 5100
 1010 NSUBDS = 0
      NRWRTN = 0
      IWFILE = NUNIT
      LRECOD = ICONTR(26,IWFILE)
      NIXOLD = ICONTR(23,IWFILE)
      NUTWTN = NUNIT
      NTHWTN = NTH
      DO 1020  LP1=1,NTH
          NODWTN(LP1,1) = NODE(LP1)
          NODWTN(LP1,2) = 2
          NODWTN(LP1,3) = 2
 1020 CONTINUE
C**** SET FIRST LEVEL DIRECTORY ADDRESS AND CLEAR NODE COUNTER
      IX = ICONTR(21,IWFILE)
      NCOUNT = 0
C
C---- 2. READ DIRECTORY AND SEARCH SUBDIRECTORY OF NODE(NCOUNT)
C
 2000 NCOUNT = NCOUNT+1
      READ(NUNIT,REC=IX)      (IBUFFR(LP1),LP1=1,LRECOD)
      IF(IBUFFR(4).GT.0)                       GO TO 2010
          IANDNM = 5
          IWRK1 = IBUFFR(4)+1
                                               GO TO 2100
C**** SEARCH SUBDIRECTORY
 2010 IANDNM = 5
      DO 2020  LP1 = 1,IBUFFR(4)
          IF(NODE(NCOUNT).EQ.IBUFFR(IANDNM))   GO TO 3000
          IANDNM = IANDNM+12
 2020 CONTINUE
      IWRK1 = IBUFFR(4)+1
C**** SUBDIRECTORY IS NOT FOUND
 2100 IF(IWRK1.LE.ICONTR(27,IWFILE))           GO TO 2110
          WRITE(6,7920)  (NODE(LP2),LP2=1,NTH)
          WRITE(6,7940) NODE(NCOUNT)
                                               GO TO 5100
 2110 IF(NCOUNT.GE.NTH)                        GO TO 2600
C
C---- 2.5 ADD SUBDIRECTORY OF NODE(NCOUNT) AND MAKE DIRECTORY OF
C         NODE(NCOUNT)
C

FILE = J3679.DPOOL2.FORT77                DATE    84/03/15(THURSDAY)     TIME 15:00:36  PAGE 00
```



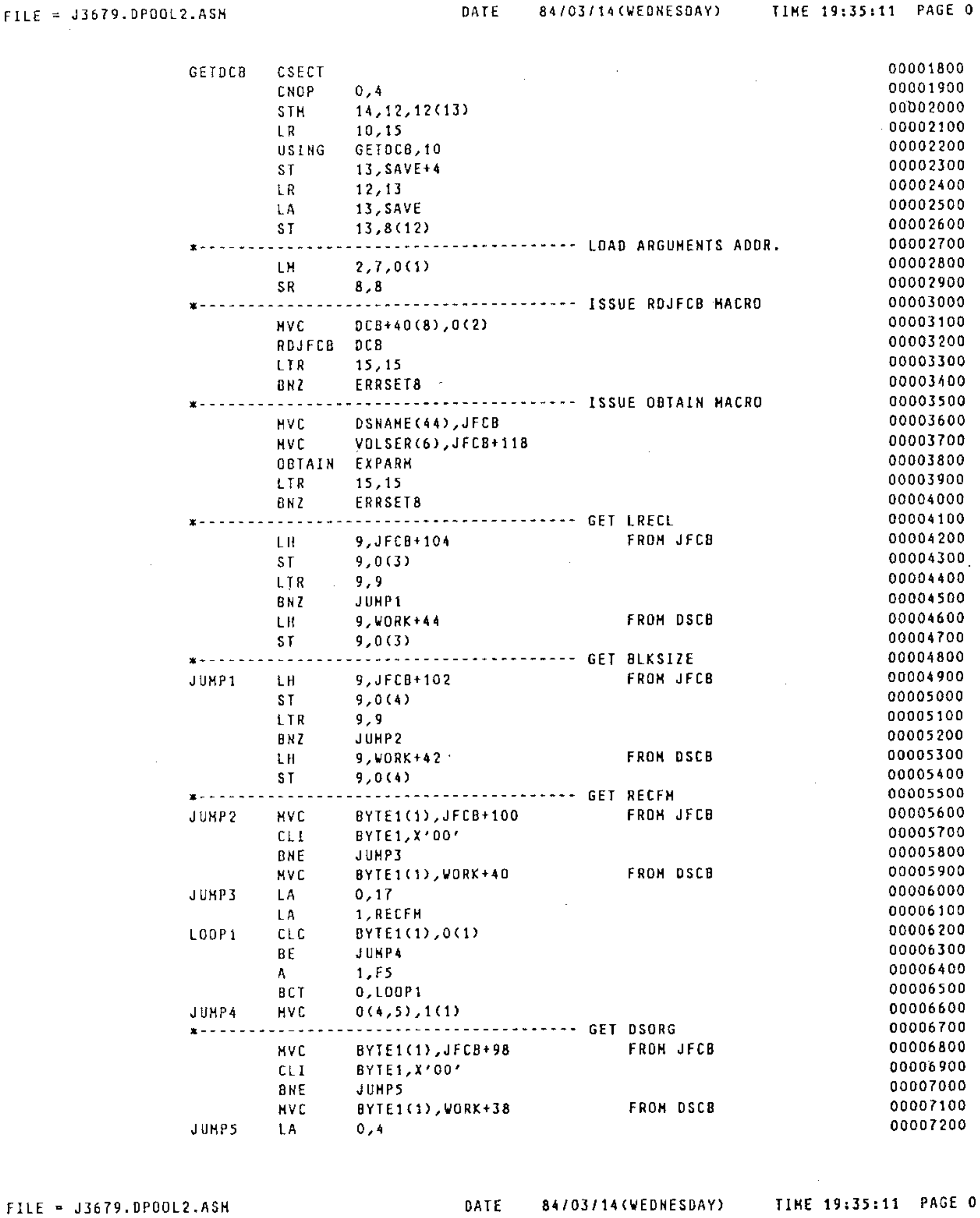
```
FILE = J3679.DPOOL2.FORT77                    DATE     84/03/15(THURSDAY)     TIME 15:00:36  PAGE 00

 2500 IBUFFR(4) = IWRK1
      NODE1 = NODE(NCOUNT)
      NDREC = 0
      NADWN = ICONTR(22,IWFILE)
      NADAT = 0
      NDASET = 0
      DO 2510  LP1=1,5
 2510 NINFOM(LP1)=0
      IWRK2=4+(IBUFFR(4)-1)*12
      DO 2520  LP1=1,12
          IWRK2=IWRK2+1
 2520     IBUFFR(IWRK2)=ISUBDR(LP1)
C     IX=IX-1
      WRITE(NUNIT,REC=IX)    (IBUFFR(LP1),LP1=1,LRECOD)
      NODWTN(NCOUNT,2) = 0
                                                 GO TO 3200
C
C---- 2.6 ADD SUBDIRECTORY AND RETURN (NODE(NCOUNT) IS LAST NODE)
C
 2600 IBUFFR(4) = IWRK1
      NODE1 = NODE(NCOUNT)
      NDREC = 0
      NADWN = 0
      NADAT = ICONTR(23,IWFILE)
      NDASET = 0
C
      DO 2610  LP1=1,5
 2610     NINFOM(LP1) = INFOM(LP1)
      IWRK2 = 4+(IBUFFR(4)-1)*12
      DO 2620  LP1=1,12
          IWRK2=IWRK2+1
 2620     IBUFFR(IWRK2) = ISUBDR(LP1)
C     IX=IX-1
      NA1=IX
      WRITE(NUNIT,REC=IX)    (IBUFFR(LP1),LP1=1,LRECOD)
      NODWTN(NCOUNT,2) = 0
      IX = NADAT
      NRETUN = 0
      NRPEMT = 1+ICONTR(28,IWFILE)+ICONTR(29,IWFILE)
     *         -(ICONTR(23,IWFILE)-1)
                                                 GO TO 5200
C
C---- 3. SUBDIRECTORY IS FOUND
C
 3000 IF(NCOUNT.GE.NTH)                          GO TO 4000
C**** NOT LAST NODE
      IWRK4=IANDNH+2
      IF(IBUFFR(IWRK4).EQ.0)                     GO TO 3100
      IX = IBUFFR(IWRK4)
      GO TO 2000
C**** THE DIRECTORY OF NODE(NCOUNT) IS NOT FOUND.
 3100 IBUFFR(IWRK4) = ICONTR(22,IWFILE)
C     IX=IX-1
      WRITE(NUNIT,REC=IX)    (IBUFFR(LP1),LP1=1,LRECOD)
      NODWTN(NCOUNT,2) = 3


FILE = J3679.DPOOL2.FORT77                    DATE     84/03/15(THURSDAY)     TIME 15:00:36  PAGE 00
FILE = J3679.DPOOL2.FORT77                 DATE     84/03/15(THURSDAY)      TIME 15:00:36  PAGE 00

C
C---- MAKE DIRECTORY OF NODE(NCOUNT)
C
 3200 IWRK3=1+ICONTR(28,IWFILE)
      IF(ICONTR(22,IWFILE).LE.IWRK3)                 GO TO 3210
C
      NTHWTN=NCOUNT
          CALL DCLEAR
          WRITE(6,7930)  (NODE(LP9),LP9=1,NTH)
                                                     GO TO 5100
C
 3210 IBUFFR(1) = NODE(NCOUNT)
      IBUFFR(2) = 0
C     IBUFFR(3) = IX-1
      IBUFFR(3) = IX
      IBUFFR(4) = 0
C
      DO 3220 LP1=5,LRECOD
 3220     IBUFFR(LP1)=0
      IX=ICONTR(22,IWFILE)
      WRITE(NUNIT,REC=IX)    (IBUFFR(LP1),LP1=1,LRECOD)
      ICONTR(22,IWFILE) = ICONTR(22,IWFILE)+1
      ICONTR(30,IWFILE) = ICONTR(30,IWFILE)+1
      NODWTN(NCOUNT,3) = 0
C     IX=IX-1
                                                     GO TO 2000
C
C---- 4. SUBDIRECTORY OF LAST NODE ALREADY EXIST.
C        UPDATE SUBDIRECTORY OR IMMEDIATELY RETURN DEPENDING ON NUPDAT
C
 4000 IWRK2 = IANDNM+3
      IF((NUPDAT.NE.0).OR .(IBUFFR(IWRK2).EQ.0))     GO TO 4010
          NRETUN = 1
                                                     GO TO 5200
C**** SAVE OLD SUBDIRECTORY TO ISDBEF(12)
 4010 IWRK3 = IANDNM - 1
      DO 4020  LP1=1,12
          IWRK3=IWRK3+1
          ISDBEF(LP1)=IBUFFR(IWRK3)
          NSDOLD(LP1)=IBUFFR(IWRK3)
 4020 CONTINUE
C**** UPDATE SUBDIRECTORY
      IWRK2=IANDNM+1
      IBUFFR(IWRK2)=0
      IWRK2=IWRK2+2
      IBUFFR(IWRK2) = ICONTR(23,IWFILE)
      IWRK2=IWRK2+1
      IBUFFR(IWRK2) = 0
      IWRK2=IWRK2+1
      IBUFFR(IWRK2) = NDATE(1)
      IWRK2=IWRK2+1
      IBUFFR(IWRK2) = NDATE(2)
      DO 4030  LP1=1,5
          IWRK2=IWRK2+1
          IBUFFR(IWRK2) = INFOM(LP1)

FILE = J3679.DPOOL2.FORT77                 DATE     84/03/15(THURSDAY)      TIME 15:00:36  PAGE 00
FILE = J3679.DPOOL2.FORT77                DATE     84/03/15(THURSDAY)     TIME 15:00:36  PAGE 00

 4030 CONTINUE
C**** SET NEW SUBDIRECTORY TO ISUBDR(12)
      IWRK5 = IANDNM-1
      DO 4040  LP1=1,12
          IWRK5=IWRK5+1
          ISUBDR(LP1) = IBUFFR(IWRK5)
 4040 CONTINUE
C
C     IX = IX-1
      NA1 = IX
      WRITE(NUNIT,REC=IX)    (IBUFFR(LP1),LP1=1,LRECOD)
C**** UPDATE CONTROL VARIABLES
      NODWTN(NCOUNT,2) = 4
      NRPEMT = 1+ICONTR(28,IWFILE)+ICONTR(29,IWFILE)-(ICONTR(23,IWFILE)
     *         -1)
      NRETUN = 0
C**** DELETE OLD DATA
      CALL DATDLT(NUNIT,NSDOLD)
C**** SET IX (WRITE STARTING POINT)
      IX = ICONTR(23,IWFILE)
C
C---- RETURN
C
 5200 RETURN
 5100 ICONTR(24,IWFILE)=0
      WRITE(NUNIT,REC=1) (ICONTR(LP1,IWFILE),LP1=1,LCONTR)
      STOP
C
C---- FORMAT
C
 7910 FORMAT(1H ,10X,'***** PSET ERROR *****'
     */15X,'THE SPECIFIED LEVEL OF THE NODE IS .LE. 0' )
 7920 FORMAT(1H ,10X,'***** PSET ERROR *****'
     */15X,'THE NUMBER OF SUBDIRECTORY OF THE SAME LEVEL NODE IS TOO LAR
     *GE'
     */15X,10(4X,A4))
 7930 FORMAT(1H ,10X,'***** PSET ERROR *****'
     */15X,'DIRECTORY AREA CAN NOT BE OBTAINED'
     */15X,10(4X,A4))
 7940 FORMAT(1H ,15X,'NODE NAME CAN NOT BE WRITTEN = ',A4)
      END

**************
** PSKIP    **
**************

      SUBROUTINE PSKIP(NUNIT,NODS)
C  SKIP OF SUB DATA SETS.
C  NODS IS NO. OF SKIPPED DATA SETS
      COMMON /DPCONT/ LCONTR,NCONTR,ICONTR(40,99),IX,NSUBDS,NDSTAT
      COMMON /DPWORK/ LBUFFR,LRECOD,IBUFFR(1000),NRECOD,NODE1,NODE2,
     *                NADWN,NADAT,NDASET,NDATE(2),NINFOM(5),NUTOLD,
     *                NTHOLD,NODOLD(10,2),NA1
      DIMENSION ICH(20),NDATA(10)
      NREC=0

FILE = J3679.DPOOL2.FORT77                DATE     84/03/15(THURSDAY)     TIME 15:00:36  PAGE 00
FILE = J3679.DPOOL2.FORT77                 DATE     84/03/15(THURSDAY)     TIME 15:00:36  PAGE 00

                    IF(NODS.EQ.0)                    GO TO 100
                    IWFILE = NUNIT
                    IXLAST=1+ICONTR(28,IWFILE)+ICONTR(29,IWFILE)
                 10 IXOLD=IX
                    READ(NUNIT,REC=IX) NODWK1,NODWK2,(ICH(LP1),LP1=1,20),
                   *               NA1WK,NSDSWK,NOA,(NDATA(N),N=1,NOA)
                    IF(NREC.EQ.0)                    GO TO 20
                    IF(NODWK1.NE.NODPRE) WRITE(6,7000) NODWK1
                 20 NODPRE=NODWK1
                    NWORD=25+NOA
                    DO 30 N=1,NOA
                 30 NWORD=NWORD+NDATA(N)
                    NRECD=NWORD/LRECOD
                    IF(MOD(NWORD,LRECOD).EQ.0)        GO TO 40
                    NRECD=NRECD+1
                 40 IX=IXOLD+NRECD
                    IF(IX.GT.IXLAST)                  GO TO 200
                    NREC=NREC+1
                    IF(NREC.LT.NODS)                  GO TO 10
                100 RETURN
                200 WRITE(6,7010) IX
                    RETURN
               7000 FORMAT('0 **** WARNING IN PSKIP ****'/
                   *        '  SKIPPED TO NEXT DATA SET (NODE NAME = ',A4,')')
               7010 FORMAT('0 **** ERROR IN PSKIP ****'/
                   *        '  DATA SET ADDRESS WAS OVERFLOWED (IX = ',I8,')')
                    END

**************
** PWEND    **
**************

                    SUBROUTINE PWEND(NUNIT)
              C
              C-------- 0. DECLARATION
              C
                    COMMON /DPCONT/ LCONTR,NCONTR,ICONTR(40,99),IX,NSUBDS,NDSTAT
                    CHARACTER FNAME*40,BLANK*4
                    DATA BLANK/'    '/
              C
              C-------- 1. SET WRITE FLAG OFF AND UPDATE CONTROL SECTION
              C
                    DO 10 N=1,37,4
                 10 FNAME(N:N+3)=BLANK
                    INQUIRE(NUNIT,NAME=FNAME)
                    IWFILE = NUNIT
                    ICONTR(24,IWFILE) = 0
                    IXUSE=ICONTR(23,IWFILE) - NDSTAT
                    IXLAST=1+ICONTR(28,IWFILE)+ICONTR(29,IWFILE)-ICONTR(23,IWFILE)+1
                    WRITE(6,7000) NUNIT,FNAME,IXUSE,IXLAST
                    WRITE(NUNIT,REC=1)     (ICONTR(LP1,IWFILE),LP1=1,LCONTR)
                    RETURN
               7000 FORMAT('0*** INFORMATION OF DATA POOL USAGE ***'/
                   *        '     LOGICAL UNIT NO.        = ',I6 /
                   *        '     DATA SET NAME           = ',A40/

FILE = J3679.DPOOL2.FORT77                 DATE     84/03/15(THURSDAY)     TIME 15:00:36  PAGE 00
FILE = J3679.DPOOL2.FORT77                   DATE    84/03/15(THURSDAY)     TIME 15:00:36  PAGE 00

                   *        '     NO. OF WRITTEN RECORDS = ',I6 /
                   *        '     REMAINS RECORDS        = ',I6 )
                    END

   **************
   ** PWSTAT   **
   **************

                    SUBROUTINE PWSTAT(NUNIT)
              C
              C-------- 0. DECLARATION
              C
                    COMMON /DPCONT/ LCONTR,NCONTR,ICONTR(40,99),IX,NSUBDS,NDSTAT
              C
              C-------- 1. READ CONTROL SECTION AND CHECK WRITE FLAG
              C
                    IWFILE = NUNIT
                    READ(NUNIT,REC=1,ERR=1110)  (ICONTR(LP1,IWFILE),LP1=1,LCONTR)
                    IF(ICONTR(24,IWFILE).NE.0)                  GO TO 2010
                                                                GO TO 2110
               1110 WRITE(6,7900)       NUNIT
                                                                GO TO 3100
              C
              C-------- 2. REWRITE CONTROL SECTION OR JOB ABORT
              C
               2010 WRITE(6,7010)       NUNIT
                                                                GO TO 3100
               2110 ICONTR(24,IWFILE) = 1
                    NDSTAT=ICONTR(23,IWFILE)
                    WRITE(NUNIT,REC=1)      (ICONTR(LP1,IWFILE),LP1=1,LCONTR)
              C
              C-------- 3. RETURN OR STOP
              C
               3000 RETURN
               3100 STOP
              C
              C-------- 4. FORMAT
              C
               7010 FORMAT(1H ,10X,'***** PWSTAT ABORT *****'
                   */15X,'WRITE FLAG IS ALREADY ON    UNIT = ',I2)
               7900 FORMAT(1H ,10X,'***** PWSTAT ERROR *****'
                   */15X,'CONTROL SECTION READ ERROR    UNIT = ',I2)
                    END

   **************
   ** SETMSG   **
   **************

                    SUBROUTINE SETMSG(RET,ERRNO,N1,N2)
              C
              C  SET ERROR SET INFORMATION IF ERROR OCCURED
              C
                    COMMON /DPEMSG/ IFLAG,NERNO
                    INTEGER RET,ERRNO
                    IFLAG=1

FILE = J3679.DPOOL2.FORT77                   DATE    84/03/15(THURSDAY)     TIME 15:00:36  PAGE 00
FILE = J3679.DPOOL2.FORT77                    DATE     84/03/15(THURSDAY)     TIME 15:00:36  PAGE 00

                    NERNO=ERRNO
                    RETURN
                    END

  **************
  ** SUBDLT   **
  **************

                    SUBROUTINE SUBDLT(NUNIT,NSDOLD)
              C
              C---- DECLARATION
              C
                    DIMENSION NSDOLD(1)
              C
              C---- DELETE DIRECTORY OF THE NODE WHICH HAS SUBDIRECTORY NSDOLD
              C
                    CALL DIRDLT(NUNIT,NSDOLD)
              C
              C---- DELETE DATA OF THE NODE WHICH HAS SUBDIRECTORY NSDOLD
              C
                    CALL DATDLT(NUNIT,NSDOLD)
                    RETURN
                    END

  **************
  ** WRTCHK   **
  **************

                    SUBROUTINE WRTCHK(NUNIT,IXOLD)
              C
              C---- DECLARATION
              C
                    COMMON /DPCONT/ LCONTR,NCONTR,ICONTR(40,99),IX,NSUBDS,NDSTAT
                    COMMON /DPWORK/ LBUFFR,LRECOD,IBUFFR(1000),NRECOD,NODE1,NODE2,
                   *                NADWN,NADAT,NDASET,NDATE(2),NINFOM(5),NUTOLD,
                   *                NTHOLD,NODOLD(10,2),NA1
                    COMMON /DPWCHK/ NRPEMT,NRWRTN,NIXOLD,NUTWTN,NTHWTN,NODWTN(10,3),
                   *                ISOBEF(12)
              C
              C---- UPDATE CONTROL SECTION
              C
                    IWFILE = NUNIT
                    ICONTR(23,IWFILE) = IX
                    ICONTR(31,IWFILE) = ICONTR(31,IWFILE)+(IX-IXOLD)
              C**** UPDATE NRWRTN AND SAVE WRITE STARTING POINT
                    NRWRTN = NRWRTN+(IX-IXOLD)
                    IXNEXT = IX
              C
              C---- UPDATE SUBDIRECTORY
              C
                    READ(NUNIT,REC=NA1)    (IBUFFR(LP1),LP1=1,LRECOD)
              C**** SEARCH SUBDIRECTORY
                    IANDNM=5
                    DO 1030 LP1=1,IBUFFR(4)
                        IF(NODE1.EQ.IBUFFR(IANDNM))                GO TO 1100

FILE = J3679.DPOOL2.FORT77                    DATE     84/03/15(THURSDAY)     TIME 15:00:36  PAGE 00
```



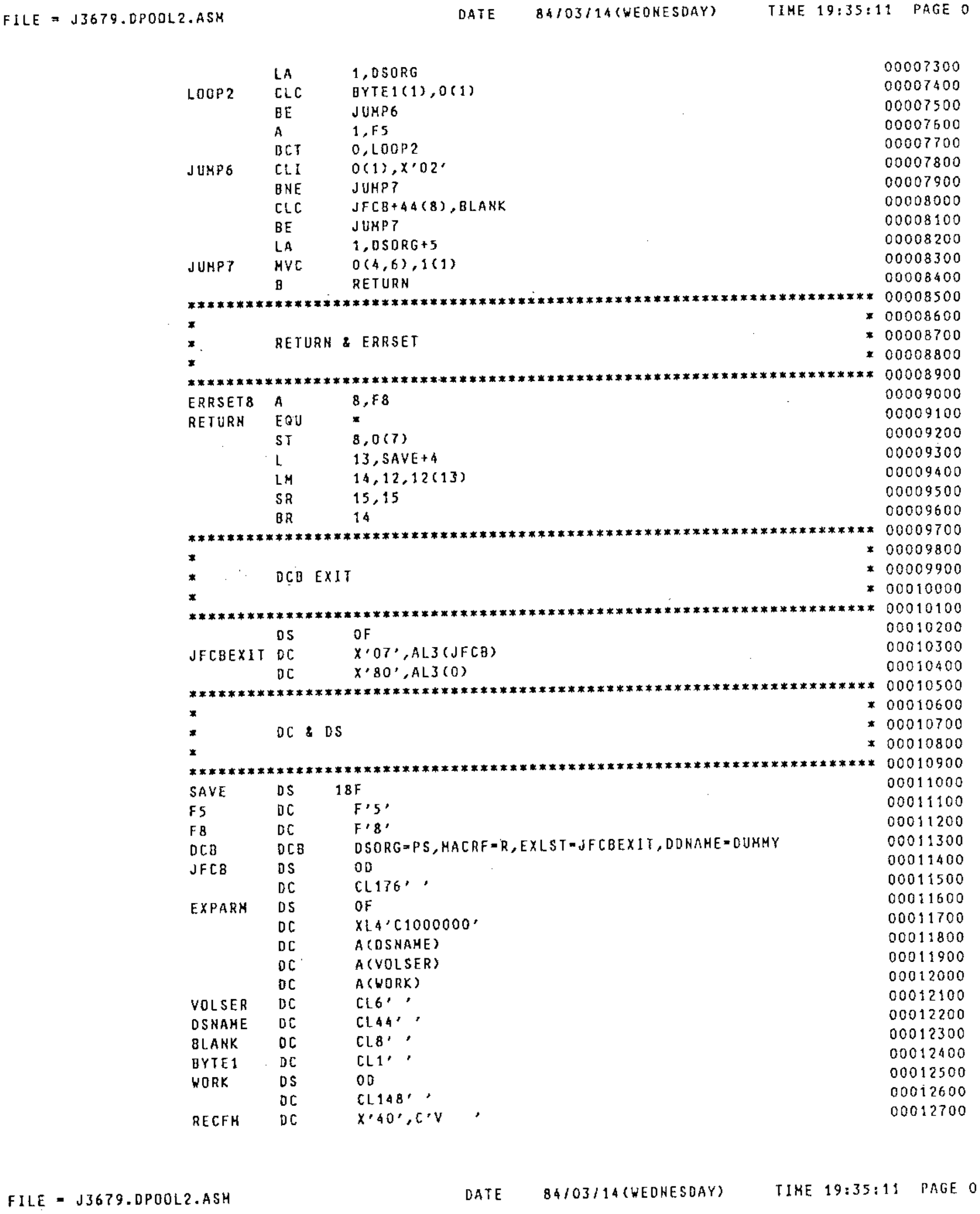
```
FILE = J3679.DPOOL2.FORT77                   DATE    84/03/15(THURSDAY)     TIME 15:00:36  PAGE 00

                IANDNM = IANDNM+12
          1030 CONTINUE
               WRITE(6,7910)
                                                        GO TO 2200
         C**** UPDATE THE NUMBER OF SUB DATA SET
          1100 IWRK = IANDNM + 1
               IBUFFR(IWRK) = NRWRTN
               IWRK = IWRK   + 3
               IBUFFR(IWRK) = NSUBDS
               NDASET = NSUBDS
               WRITE(NUNIT,REC=NA1)   (IBUFFR(LP1),LP1=1,LRECOD)
               IX = IXNEXT
         C
          2100 RETURN
          2200 STOP
         C
          7910 FORMAT(1H ,10X,'***** WRTCHK ERROR *****'
              */15X,'SUB-DIRECTORY CAN NOT BE FOUND')
         C
               END
```

```
FILE = J3679.DPOOL2.ASM                    DATE    84/03/14(WEDNESDAY)     TIME 19:35:11  PAGE 0



                                  DIRECTORY LIST OF J3679.DPOOL2.ASM

                              **************  ***************   ***************
                               MEMBER NAME      PAGE NO.          NO. OF CARDS
                              **************  ***************   ***************

                               GETDCB             0007                150
```

```
**************
** GETDCB   **
**************

               ************************************************************************ 00000100
               *                                                                      * 00000200
               *        'GETDCB'                                                      * 00000300
               *                                                                      * 00000400
               *             CALL  GETDCB ( DDNAME,LRECL,BLKSIZE,RECFM,DSORG,RCD )    * 00000500
               *                                                                      * 00000600
               *                   DDNAME       C*8   INPUT                           * 00000700
               *                   LRECL        I*4   OUTPUT                          * 00000800
               *                   BLKSIZE      I*4   OUTPUT                          * 00000900
               *                   RECFM        C*4   OUTPUT                          * 00001000
               *                                        IF 'XXXX'      UNKNOWN        * 00001100
               *                   DSORG        C*4   OUTPUT                          * 00001200
               *                                        IF 'XXXX'      UNKNOWN        * 00001300
               *                   RCD          I*4   OUTPUT                          * 00001400
               *                                        IF RCD=0       NORMAL END     * 00001500
               *                                        IF RCD=8       DD MISSING     * 00001600
               ************************************************************************ 00001700

FILE = J3679.DPOOL2.ASM                    DATE    84/03/14(WEDNESDAY)     TIME 19:35:11  PAGE 0
```

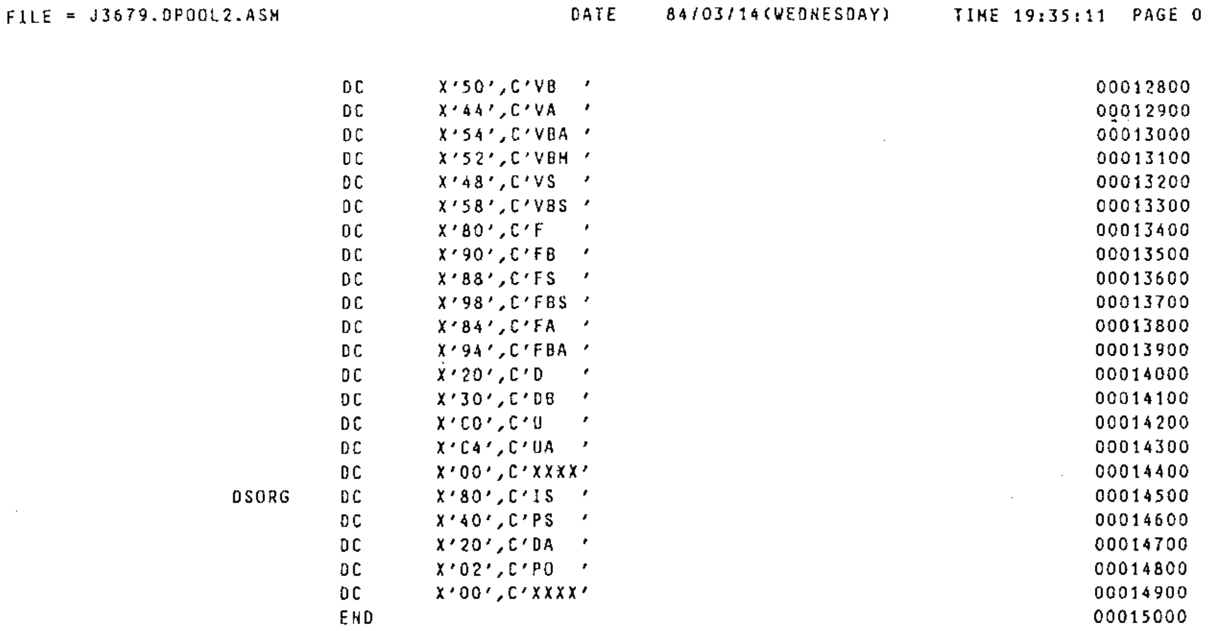


```
FILE = J3679.DPOOL2.ASM                    DATE    84/03/14(WEDNESDAY)    TIME 19:35:11  PAGE 0

                GETDCB   CSECT                                                               00001800
                         CNOP     0,4                                                        00001900
                         STM      14,12,12(13)                                               00002000
                         LR       10,15                                                      00002100
                         USING    GETDCB,10                                                  00002200
                         ST       13,SAVE+4                                                  00002300
                         LR       12,13                                                      00002400
                         LA       13,SAVE                                                    00002500
                         ST       13,8(12)                                                   00002600
                *---------------------------------------- LOAD ARGUMENTS ADDR.               00002700
                         LM       2,7,0(1)                                                   00002800
                         SR       8,8                                                        00002900
                *---------------------------------------- ISSUE RDJFCB MACRO                 00003000
                         MVC      DCB+40(8),0(2)                                             00003100
                         RDJFCB   DCB                                                        00003200
                         LTR      15,15                                                      00003300
                         BNZ      ERRSET8                                                    00003400
                *---------------------------------------- ISSUE OBTAIN MACRO                 00003500
                         MVC      DSNAME(44),JFCB                                            00003600
                         MVC      VOLSER(6),JFCB+118                                         00003700
                         OBTAIN   EXPARM                                                     00003800
                         LTR      15,15                                                      00003900
                         BNZ      ERRSET8                                                    00004000
                *---------------------------------------- GET LRECL                          00004100
                         LH       9,JFCB+104                  FROM JFCB                      00004200
                         ST       9,0(3)                                                     00004300
                         LTR      9,9                                                        00004400
                         BNZ      JUMP1                                                      00004500
                         LH       9,WORK+44                   FROM DSCB                      00004600
                         ST       9,0(3)                                                     00004700
                *---------------------------------------- GET BLKSIZE                        00004800
                JUMP1    LH       9,JFCB+102                  FROM JFCB                      00004900
                         ST       9,0(4)                                                     00005000
                         LTR      9,9                                                        00005100
                         BNZ      JUMP2                                                      00005200
                         LH       9,WORK+42                   FROM DSCB                      00005300
                         ST       9,0(4)                                                     00005400
                *---------------------------------------- GET RECFM                          00005500
                JUMP2    MVC      BYTE1(1),JFCB+100           FROM JFCB                      00005600
                         CLI      BYTE1,X'00'                                                00005700
                         BNE      JUMP3                                                      00005800
                         MVC      BYTE1(1),WORK+40            FROM DSCB                      00005900
                JUMP3    LA       0,17                                                       00006000
                         LA       1,RECFM                                                    00006100
                LOOP1    CLC      BYTE1(1),0(1)                                              00006200
                         BE       JUMP4                                                      00006300
                         A        1,F5                                                       00006400
                         BCT      0,LOOP1                                                    00006500
                JUMP4    MVC      0(4,5),1(1)                                                00006600
                *---------------------------------------- GET DSORG                          00006700
                         MVC      BYTE1(1),JFCB+98            FROM JFCB                      00006800
                         CLI      BYTE1,X'00'                                                00006900
                         BNE      JUMP5                                                      00007000
                         MVC      BYTE1(1),WORK+38            FROM DSCB                      00007100
                JUMP5    LA       0,4                                                        00007200

FILE = J3679.DPOOL2.ASM                    DATE    84/03/14(WEDNESDAY)    TIME 19:35:11  PAGE 0
```



```
FILE = J3679.DPOOL2.ASM                 DATE    84/03/14(WEDNESDAY)     TIME 19:35:11  PAGE 0

                    LA      1,DSORG                                                      00007300
                    LOOP2    CLC     BYTE1(1),0(1)                                       00007400
```

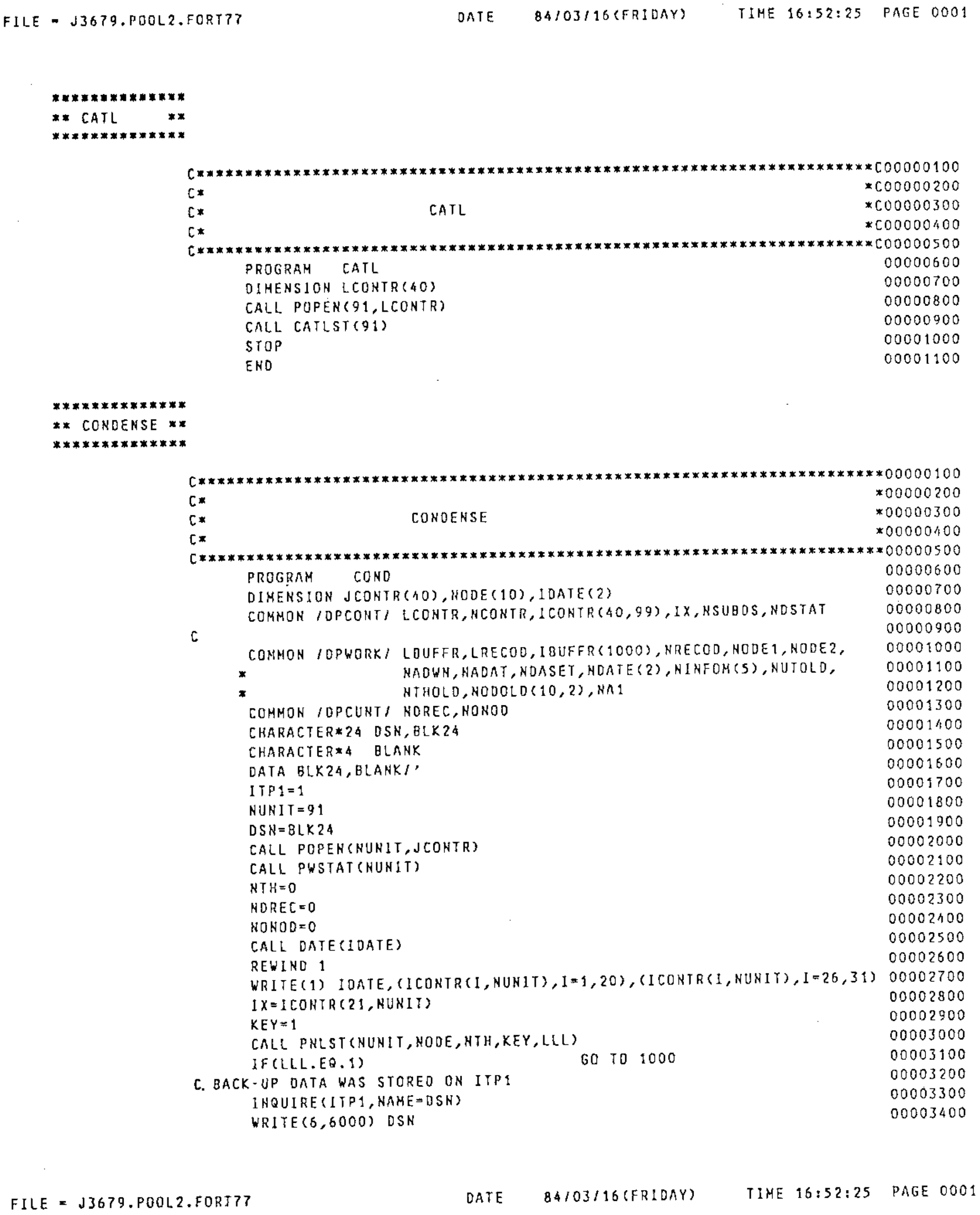
```
         LA       1,DSORG                                                00007300
LOOP2    CLC      BYTE1(1),0(1)                                          00007400
         BE       JUMP6                                                  00007500
         A        1,F5                                                   00007600
         BCT      0,LOOP2                                                00007700
JUMP6    CLI      0(1),X'02'                                             00007800
         BNE      JUMP7                                                  00007900
         CLC      JFCB+44(8),BLANK                                       00008000
         BE       JUMP7                                                  00008100
         LA       1,DSORG+5                                              00008200
JUMP7    MVC      0(4,6),1(1)                                            00008300
         B        RETURN                                                 00008400
*********************************************************************** 00008500
*                                                                     * 00008600
*        RETURN & ERRSET                                              * 00008700
*                                                                     * 00008800
*********************************************************************** 00008900
ERRSET8  A        8,F8                                                   00009000
RETURN   EQU      *                                                      00009100
         ST       8,0(7)                                                 00009200
         L        13,SAVE+4                                              00009300
         LM       14,12,12(13)                                           00009400
         SR       15,15                                                  00009500
         BR       14                                                     00009600
*********************************************************************** 00009700
*                                                                     * 00009800
*        DCB EXIT                                                     * 00009900
*                                                                     * 00010000
*********************************************************************** 00010100
         DS       0F                                                     00010200
JFCBEXIT DC       X'07',AL3(JFCB)                                        00010300
         DC       X'80',AL3(0)                                           00010400
*********************************************************************** 00010500
*                                                                     * 00010600
*        DC & DS                                                      * 00010700
*                                                                     * 00010800
*********************************************************************** 00010900
SAVE     DS     18F                                                      00011000
F5       DC       F'5'                                                   00011100
F8       DC       F'8'                                                   00011200
DCB      DCB      DSORG=PS,MACRF=R,EXLST=JFCBEXIT,DDNAME=DUMMY           00011300
JFCB     DS       0D                                                     00011400
         DC       CL176' '                                               00011500
EXPARM   DS       0F                                                     00011600
         DC       XL4'C1000000'                                          00011700
         DC       A(DSNAME)                                              00011800
         DC       A(VOLSER)                                              00011900
         DC       A(WORK)                                                00012000
VOLSER   DC       CL6' '                                                 00012100
DSNAME   DC       CL44' '                                                00012200
BLANK    DC       CL8' '                                                 00012300
BYTE1    DC       CL1' '                                                 00012400
WORK     DS       0D                                                     00012500
         DC       CL148' '                                               00012600
RECFM    DC       X'40',C'V   '                                          00012700

FILE = J3679.DPOOL2.ASM                 DATE    84/03/14(WEDNESDAY)     TIME 19:35:11  PAGE 0
```



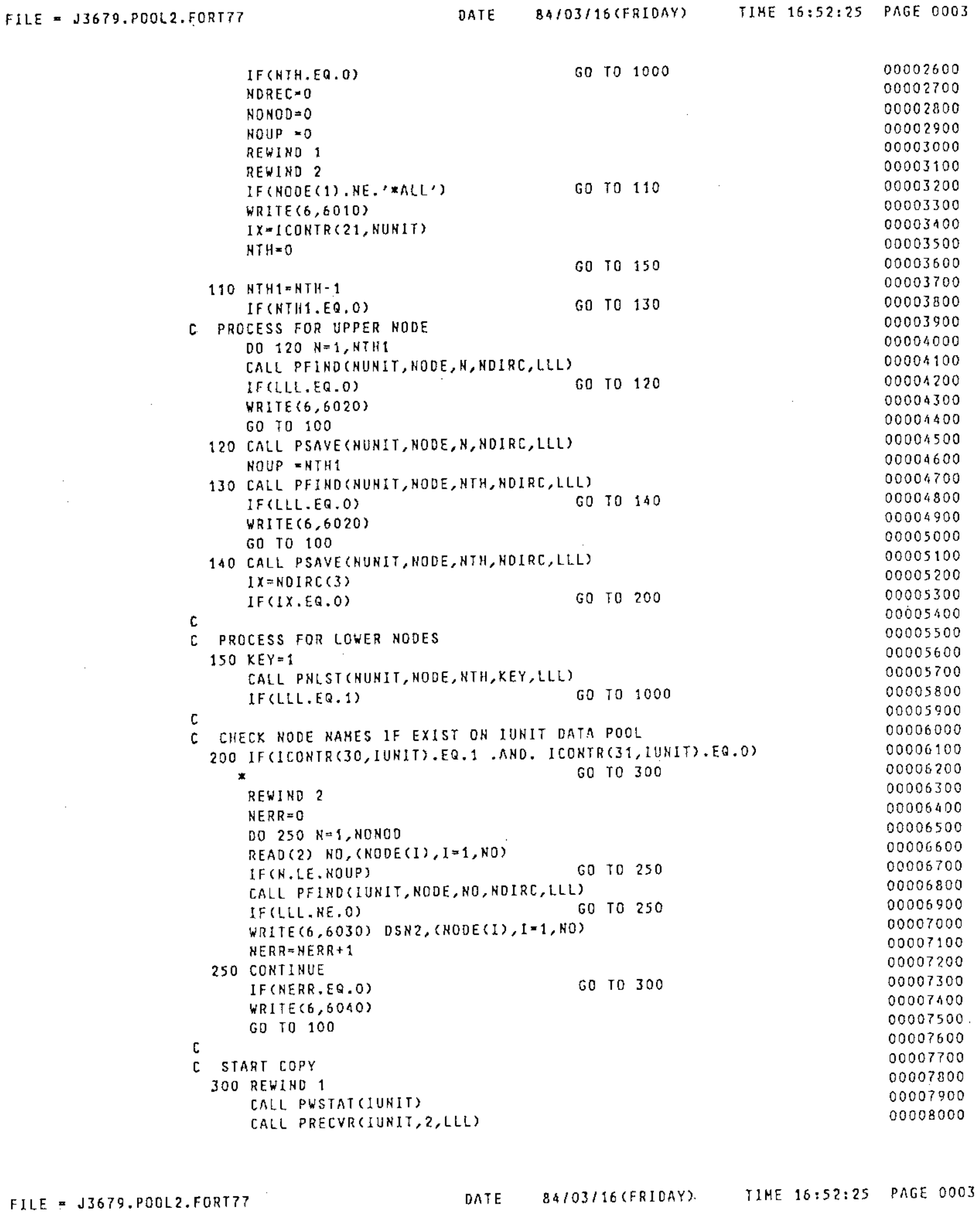
```
FILE = J3679.DPOOL2.ASM                    DATE    84/03/14(WEDNESDAY)     TIME 19:35:11  PAGE 0

                    DC      X'50',C'VB  '                                                 00012800
                    DC      X'44',C'VA  '                                                 00012900
                    DC      X'54',C'VBA '                                                 00013000
                    DC      X'52',C'VBH '                                                 00013100
                    DC      X'48',C'VS  '                                                 00013200
                    DC      X'58',C'VBS '                                                 00013300
                    DC      X'80',C'F   '                                                 00013400
                    DC      X'90',C'FB  '                                                 00013500
                    DC      X'88',C'FS  '                                                 00013600
                    DC      X'98',C'FBS '                                                 00013700
                    DC      X'84',C'FA  '                                                 00013800
                    DC      X'94',C'FBA '                                                 00013900
                    DC      X'20',C'D   '                                                 00014000
                    DC      X'30',C'DB  '                                                 00014100
                    DC      X'C0',C'U   '                                                 00014200
                    DC      X'C4',C'UA  '                                                 00014300
                    DC      X'00',C'XXXX'                                                 00014400
           DSORG    DC      X'80',C'IS  '                                                 00014500
                    DC      X'40',C'PS  '                                                 00014600
                    DC      X'20',C'DA  '                                                 00014700
                    DC      X'02',C'PO  '                                                 00014800
                    DC      X'00',C'XXXX'                                                 00014900
                    END                                                                   00015000
```

Appendix C Program List of POOL

```
                        DIRECTORY LIST OF J3679.POOL2.FORT77

              **************   ***************     ***************
               MEMBER NAME      PAGE NO.            NO. OF CARDS
              **************   ***************     ***************

(NO.=001)      CATL               0001                   11
(NO.=002)      CONDENSE           0001                   59
(NO.=003)      COPY               0002                   97
(NO.=004)      DELETE             0004                   22
(NO.=005)      FLAG               0004                   22
(NO.=006)      HELP               0005                   27
(NO.=007)      INIT               0005                   26
(NO.=008)      LIST               0006                   62
(NO.=009)      MEND               0007                  348
(NO.=010)      MTCOPY             0014                   44
(NO.=011)      MTSAVE             0014                   54
(NO.=012)      RENAME             0016                   91
(NO.=013)      TREE               0017                   41
                                                   -----------------
                                                        904 CARDS
```



```
FILE - J3679.POOL2.FORT77                    DATE     84/03/16(FRIDAY)      TIME 16:52:25  PAGE 0001


    **************
    ** CATL     **
    **************

                C**********************************************************************C00000100
                C*                                                                    *C00000200
                C*                           CATL                                     *C00000300
                C*                                                                    *C00000400
                C**********************************************************************C00000500
                      PROGRAM   CATL                                                    00000600
                      DIMENSION LCONTR(40)                                              00000700
                      CALL POPEN(91,LCONTR)                                             00000800
                      CALL CATLST(91)                                                   00000900
                      STOP                                                              00001000
                      END                                                               00001100

    **************
    ** CONDENSE **
    **************

                C***********************************************************************00000100
                C*                                                                     *00000200
                C*                         CONDENSE                                    *00000300
                C*                                                                     *00000400
                C***********************************************************************00000500
                      PROGRAM    COND                                                   00000600
                      DIMENSION JCONTR(40),NODE(10),IDATE(2)                            00000700
                      COMMON /DPCONT/ LCONTR,NCONTR,ICONTR(40,99),IX,NSUBDS,NDSTAT      00000800
                C                                                                       00000900
                      COMMON /DPWORK/ LBUFFR,LRECOD,IBUFFR(1000),NRECOD,NODE1,NODE2,    00001000
                     *                NADWN,NADAT,NDASET,NDATE(2),NINFOM(5),NUTOLD,     00001100
                     *                NTHOLD,NODOLD(10,2),NA1                           00001200
                      COMMON /DPCUNT/ NDREC,NONOD                                       00001300
                      CHARACTER*24 DSN,BLK24                                            00001400
                      CHARACTER*4  BLANK                                                00001500
                      DATA BLK24,BLANK/'                                                00001600
                      ITP1=1                                                            00001700
                      NUNIT=91                                                          00001800
                      DSN=BLK24                                                         00001900
                      CALL POPEN(NUNIT,JCONTR)                                          00002000
                      CALL PWSTAT(NUNIT)                                                00002100
                      NTH=0                                                             00002200
                      NDREC=0                                                           00002300
                      NONOD=0                                                           00002400
                      CALL DATE(IDATE)                                                  00002500
                      REWIND 1                                                          00002600
                      WRITE(1) IDATE,(ICONTR(I,NUNIT),I=1,20),(ICONTR(I,NUNIT),I=26,31) 00002700
                      IX=ICONTR(21,NUNIT)                                               00002800
                      KEY=1                                                             00002900
                      CALL PNLST(NUNIT,NODE,NTH,KEY,LLL)                                00003000
                      IF(LLL.EQ.1)                        GO TO 1000                    00003100
                C. BACK-UP DATA WAS STORED ON ITP1                                      00003200
                      INQUIRE(ITP1,NAME=DSN)                                            00003300
                      WRITE(6,6000) DSN                                                 00003400


FILE = J3679.POOL2.FORT77                    DATE     84/03/16(FRIDAY)      TIME 16:52:25  PAGE 0001
FILE = J3679.POOL2.FORT77                 DATE    84/03/16(FRIDAY)     TIME 16:52:25  PAGE 0002

C RESET CONTROL SECTION
      ICONTR(22,NUNIT)=3
      ICONTR(23,NUNIT)=ICONTR(28,NUNIT)+2
      ICONTR(24,NUNIT)=1
      ICONTR(30,NUNIT)=1
      ICONTR(31,NUNIT)=0
      NUTOLD=0
      NTHOLD=0
      NDSTAT=ICONTR(23,NUNIT)
      WRITE(NUNIT,REC=1) (ICONTR(LP1,NUNIT),LP1=1,LCONTR)
      READ(BLANK,'(A4)') IBUFFR(1)
      READ(BLANK,'(A4)') IBUFFR(2)
      IBUFFR(3)=0
      IBUFFR(4)=0
      WRITE(NUNIT,REC=2) (IBUFFR(LP1),LP1=1,4)
C LOAD BACK-UP DATA
      REWIND 1
      READ(1)
      CALL PRECVR(NUNIT,2,LLL)
      CALL PWEND(NUNIT)
C
 1000 STOP
C
 6000 FORMAT(' BACK-UP DATA SET ',A24,'WAS CREATED')
      END

**************
** COPY     **
**************

C***********************************************************************
C*                                                                     *
C*                        COPY                                         *
C*                                                                     *
C***********************************************************************
      PROGRAM   COPY
      DIMENSION JCONTR(40),NDIRC(12)
      COMMON /DPCONT/ LCONTR,NCONTR,ICONTR(40,99),IX,NSUBDS,NDSTAT
C
      COMMON /DPWORK/ LBUFFR,LRECOD,IBUFFR(1000),NRECOD,NODE1,NODE2,
     *                NADWN,NADAT,NDASET,NDATE(2),NINFOM(5),NUTOLD,
     *                NTHOLD,NODOLD(10,2),NA1
      COMMON /DPCUNT/ NDREC,NONOD
      CHARACTER*4 NODE(10)
      CHARACTER*24 DSN2,BLK24
      DATA BLK24/'
C
      DSN2=BLK24
      NUNIT=91
      IUNIT=92
      CALL POPEN(NUNIT,JCONTR)
      CALL POPEN(IUNIT,JCONTR)
      INQUIRE(IUNIT,NAME=DSN2)
   50 WRITE(6,6000)
  100 CALL NODINP(NTH,NODE)

FILE = J3679.POOL2.FORT77                 DATE    84/03/16(FRIDAY)     TIME 16:52:25  PAGE 0002
```



```
FILE = J3679.POOL2.FORT77                    DATE    84/03/16(FRIDAY)     TIME 16:52:25  PAGE 0003

            IF(NTH.EQ.0)                     GO TO 1000
            NDREC=0
            NONOD=0
            NOUP =0
            REWIND 1
            REWIND 2
            IF(NODE(1).NE.'*ALL')            GO TO 110
            WRITE(6,6010)
            IX=ICONTR(21,NUNIT)
            NTH=0
                                             GO TO 150
        110 NTH1=NTH-1
            IF(NTH1.EQ.0)                    GO TO 130
      C  PROCESS FOR UPPER NODE
            DO 120 N=1,NTH1
            CALL PFIND(NUNIT,NODE,N,NDIRC,LLL)
            IF(LLL.EQ.0)                     GO TO 120
            WRITE(6,6020)
            GO TO 100
        120 CALL PSAVE(NUNIT,NODE,N,NDIRC,LLL)
            NOUP =NTH1
        130 CALL PFIND(NUNIT,NODE,NTH,NDIRC,LLL)
            IF(LLL.EQ.0)                     GO TO 140
            WRITE(6,6020)
            GO TO 100
        140 CALL PSAVE(NUNIT,NODE,NTH,NDIRC,LLL)
            IX=NDIRC(3)
            IF(IX.EQ.0)                      GO TO 200
      C
      C  PROCESS FOR LOWER NODES
        150 KEY=1
            CALL PNLST(NUNIT,NODE,NTH,KEY,LLL)
            IF(LLL.EQ.1)                     GO TO 1000
      C
      C  CHECK NODE NAMES IF EXIST ON IUNIT DATA POOL
        200 IF(ICONTR(30,IUNIT).EQ.1 .AND. ICONTR(31,IUNIT).EQ.0)
           *                                 GO TO 300
            REWIND 2
            NERR=0
            DO 250 N=1,NONOD
            READ(2) NO,(NODE(I),I=1,NO)
            IF(N.LE.NOUP)                    GO TO 250
            CALL PFIND(IUNIT,NODE,NO,NDIRC,LLL)
            IF(LLL.NE.0)                     GO TO 250
            WRITE(6,6030) DSN2,(NODE(I),I=1,NO)
            NERR=NERR+1
        250 CONTINUE
            IF(NERR.EQ.0)                    GO TO 300
            WRITE(6,6040)
            GO TO 100
      C
      C  START COPY
        300 REWIND 1
            CALL PWSTAT(IUNIT)
            CALL PRECVR(IUNIT,2,LLL)

FILE = J3679.POOL2.FORT77                    DATE    84/03/16(FRIDAY)     TIME 16:52:25  PAGE 0003
```

```
      CALL PWEND(IUNIT)
C
      WRITE(6,6050)
      IF(NTH.EQ.0)                      GO TO 1000
      WRITE(6,6060)
      GO TO 100
 1000 STOP
 6000 FORMAT(' ENTER NODE NAME. IF *ALL IS ENTERD, ALL DATA IS COPIED')
 6010 FORMAT(' ALL DATA IS COPIED')
 6020 FORMAT(' INPUT NODE NAME IS NOT FOUND.'/' PLEASE RE-ENTER NODE NAM
   *E')
 6030 FORMAT(' EXIST SAME NODE NAME IN A DATA POOL ',A24
     *       /'  NODE NAME = ',A4,9('.',A4))
 6040 FORMAT(' COPY IS NOT EXECUTED. PLEASE RE-ENTER')
 6050 FORMAT(' DATA COPY WAS FINISHED SUCCESSFULLY')
 6060 FORMAT(' ENTER NEXT NODE NAME')
      END
```

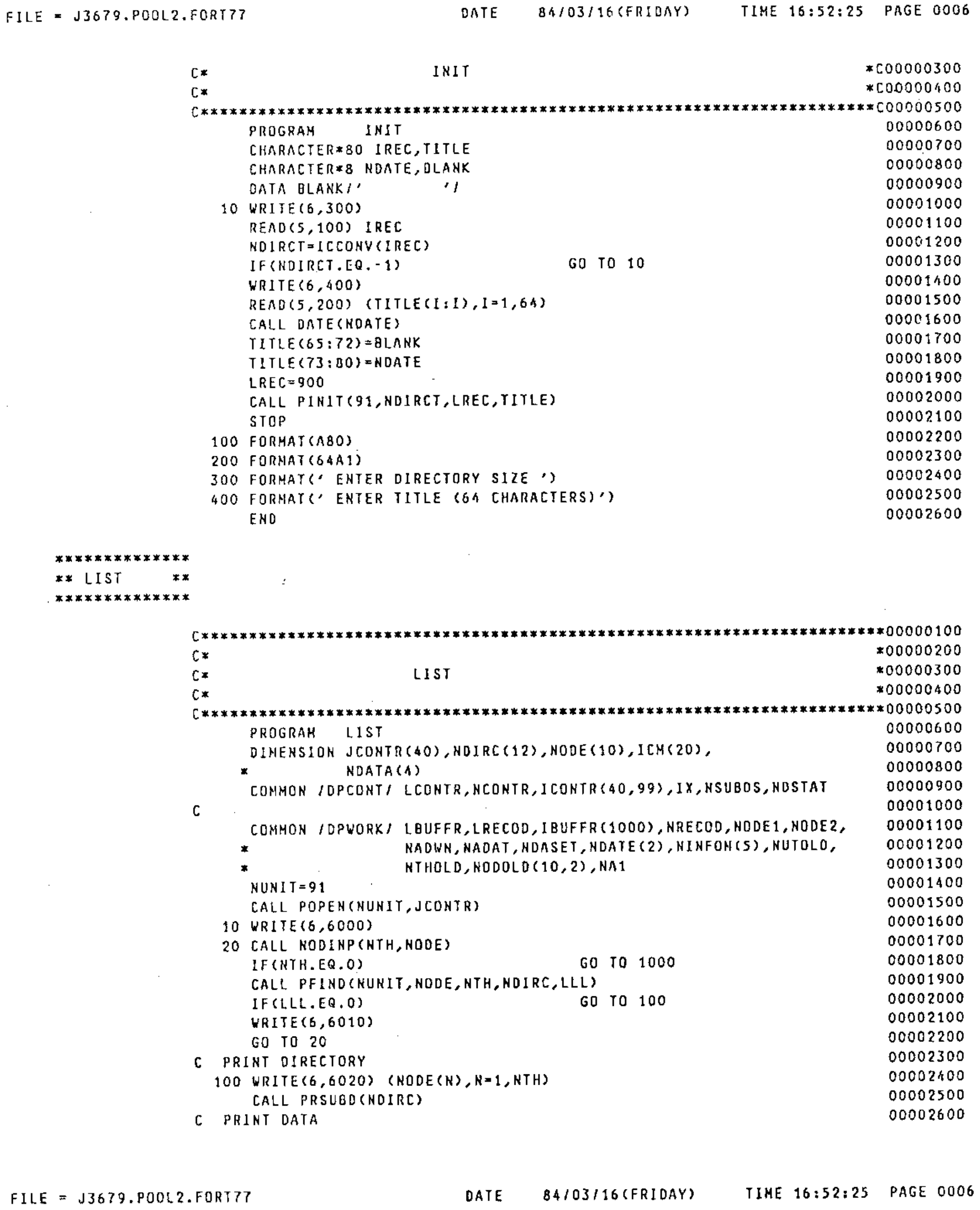
```
**************
** DELETE   **
**************
```

```
C***********************************************************************
C*                                                                     *
C*                     DELETE                                          *
C*                                                                     *
C***********************************************************************
      PROGRAM   DELETE
      DIMENSION NODE(10),LCONTR(40)
      CALL POPEN(91,LCONTR)
    1 CONTINUE
      WRITE(6,300)
      CALL NODINP(NTH,NODE)
      IF(NTH.EQ.0) GO TO 500
      CALL PDELT(91,NODE,NTH,NRETRN)
      IF(NRETRN.EQ.0) WRITE(6,200)  (NODE(I),I=1,NTH)
      IF(NRETRN.NE.0) WRITE(6,250)  (NODE(I),I=1,NTH)
      GO TO 1
  500 CONTINUE
      STOP
  200 FORMAT(' NORMAL RETURN  ***  NODE NAME = ',A4,9('.',A4))
  250 FORMAT(' ABNORMAL RETURN  ***  NODE NAME = ',A4,9('.',A4))
  300 FORMAT(' ENTER NODE NAME.')
      END
```

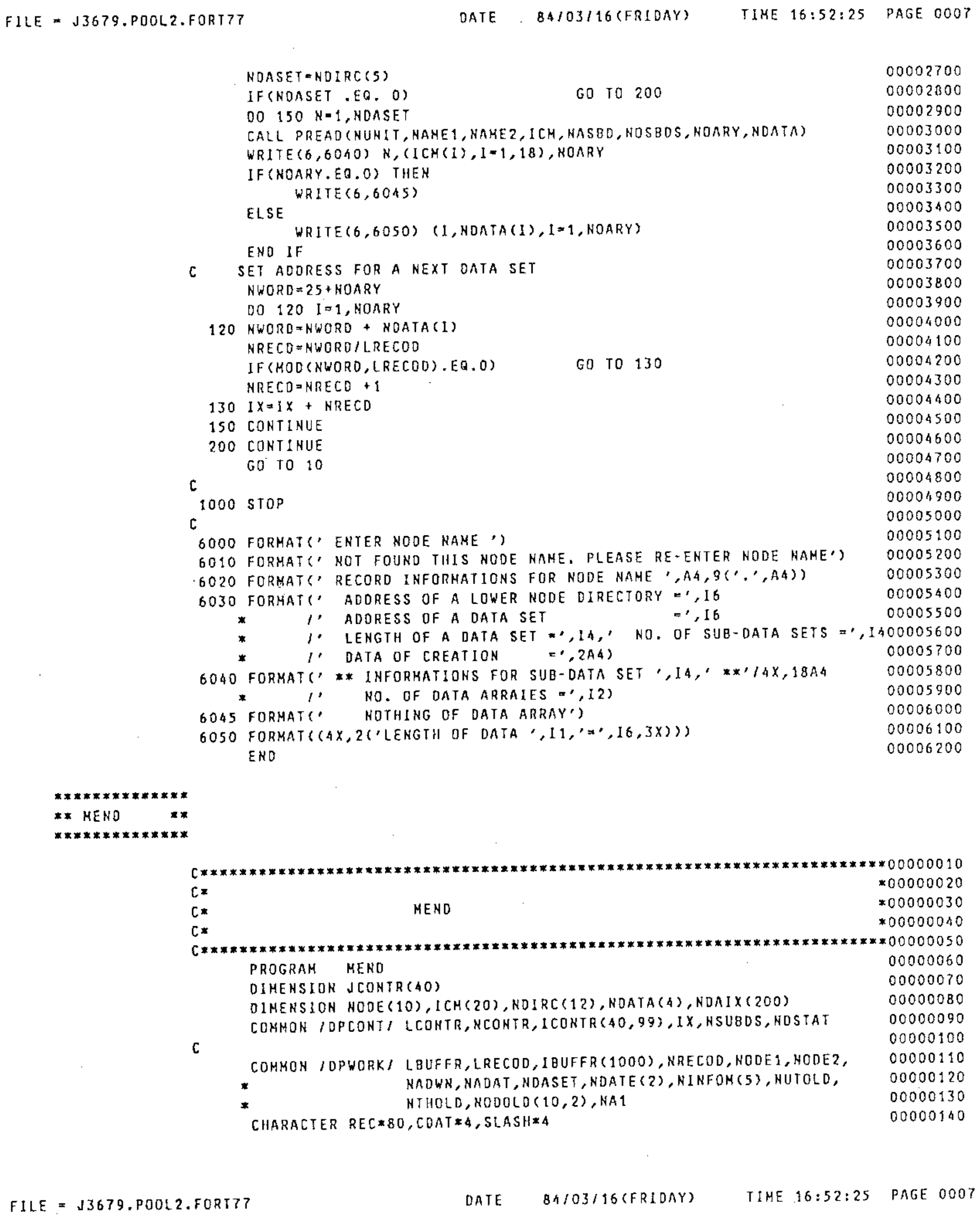
```
**************
** FLAG     **
**************
```

```
C***********************************************************************
C*                                                                     *
C*                       FLAG                                          *
C*                                                                     *
C***********************************************************************
      PROGRAM   FLAG
```

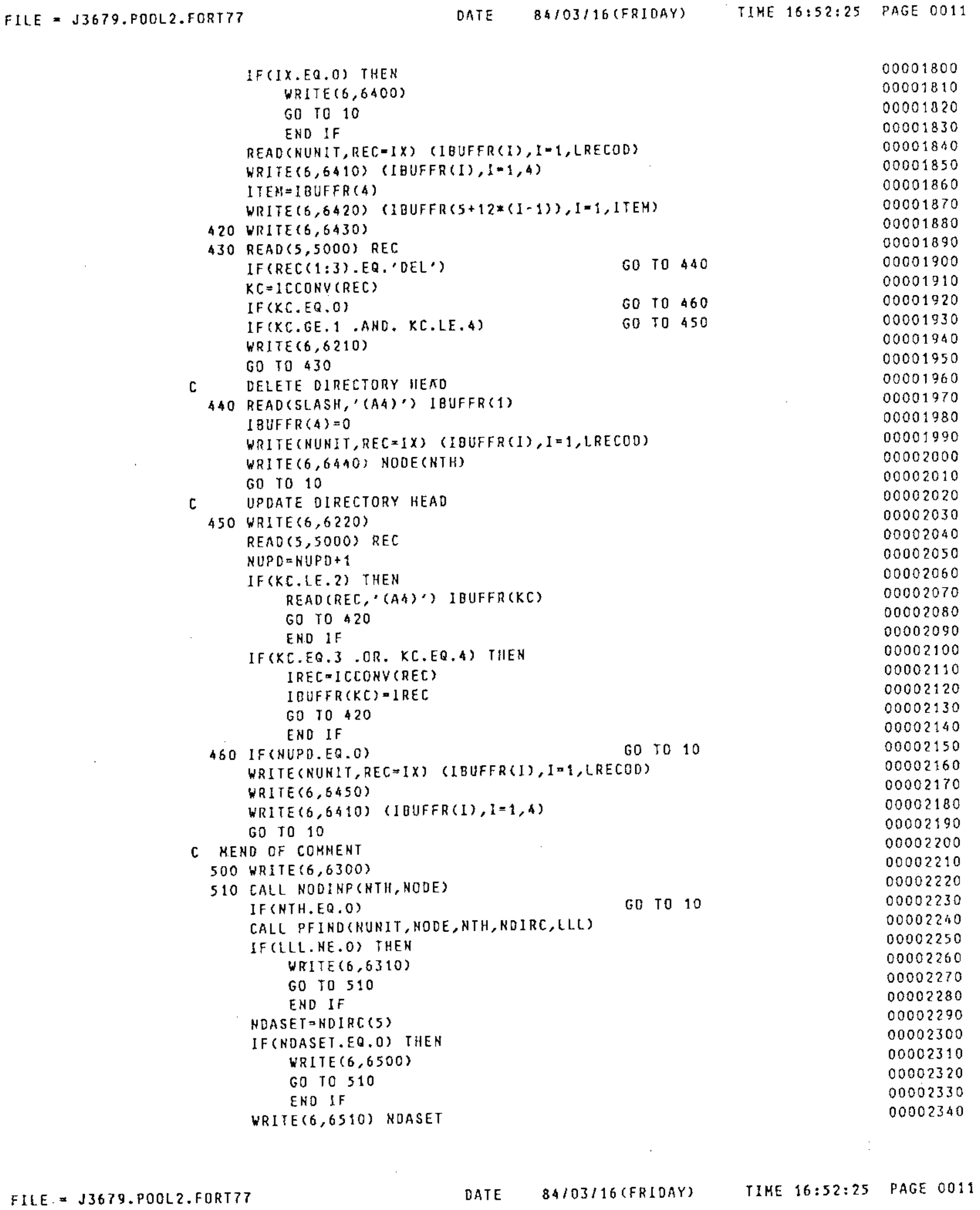
```
FILE = J3679.POOL2.FORT77                    DATE    84/03/16(FRIDAY)     TIME 16:52:25  PAGE 0005

                    DIMENSION JCONTR(40)
                    COMMON /DPCONT/ LCONTR,NCONTR,ICONTR(40,99),IX,NSUBDS,NDSTAT
                    COMMON /DPWORK/ LBUFFR,LRECOD,IBUFFR(1000),NRECOD,NODE1,NODE2,
                   *                NADWN,NADAT,NDASET,NDATE(2),NINFOM(5),NUTOLD,
                   *                NTHOLD,NODOLD(10,2),NA1
              C
                    NUNIT=91
                    CALL POPEN(NUNIT,JCONTR)
                    WRITE(6,100) ICONTR(24,NUNIT)
                    ICONTR(24,NUNIT)=0
                    WRITE(6,200)
                    WRITE(NUNIT,REC=1)  (ICONTR(I,NUNIT),I=1,LRECOD)
                    STOP
                100 FORMAT('  CURRENT STATUS OF WRITE FLAG = ',I3)
                200 FORMAT('  NOW CHANGE WRITE FLAG TO 0')
                    END

**************
** HELP     **
**************

              C**********************************************************************
              C                                                                     *
              C                  HELP                                               *
              C                                                                     *
              C**********************************************************************
                    PROGRAM HELP
              C
                    WRITE(6,6000)
              C
                    STOP
              C
               6000 FORMAT(' COMMAND                CONTENTS'/
                   *       ' CATL        PRINT OF CONTROL AND DIRECTORY SECTION'/
                   *       ' CONDENSE    CONDENSE OF A DATA POOL'/
                   *       ' COPY        COPY OF A NODE DATA'/
                   *       ' DELETE      DELETE OF A NODE DATA'/
                   *       ' FLAG        CHANGE OF A WRITE FLAG'/
                   *       ' INIT        INITIALIZATION OF A DATA POOL'/
                   *       ' LIST        LISTING OF A NODE DATA (SUB-DIRECTORY AND FORM
                   *OF DATA ARRAIES)'/
                   *       ' MEND        MENDING OF A CONTROL, DIRECTORY AND DATA COMMEN
                   *T'/    ' MTCOPY      LOAD OF A BACK-UP TAPE TO A DATA POOL'/
                   *       ' MTSAVE      MAKING OF A BACK-UP TAPE'/
                   *       ' RENAME      RENAME OF A NODE'/
                   *       ' TREE        PRINT OF ALL NODE NAMES IN A DATA POOL BY A TRE
                   *E STRUCTURE')
                    END

**************
** INIT     **
**************

              C***********************************************************************C
              C*                                                                     *C

FILE = J3679.POOL2.FORT77                    DATE    84/03/16(FRIDAY)     TIME 16:52:25  PAGE 0005
```



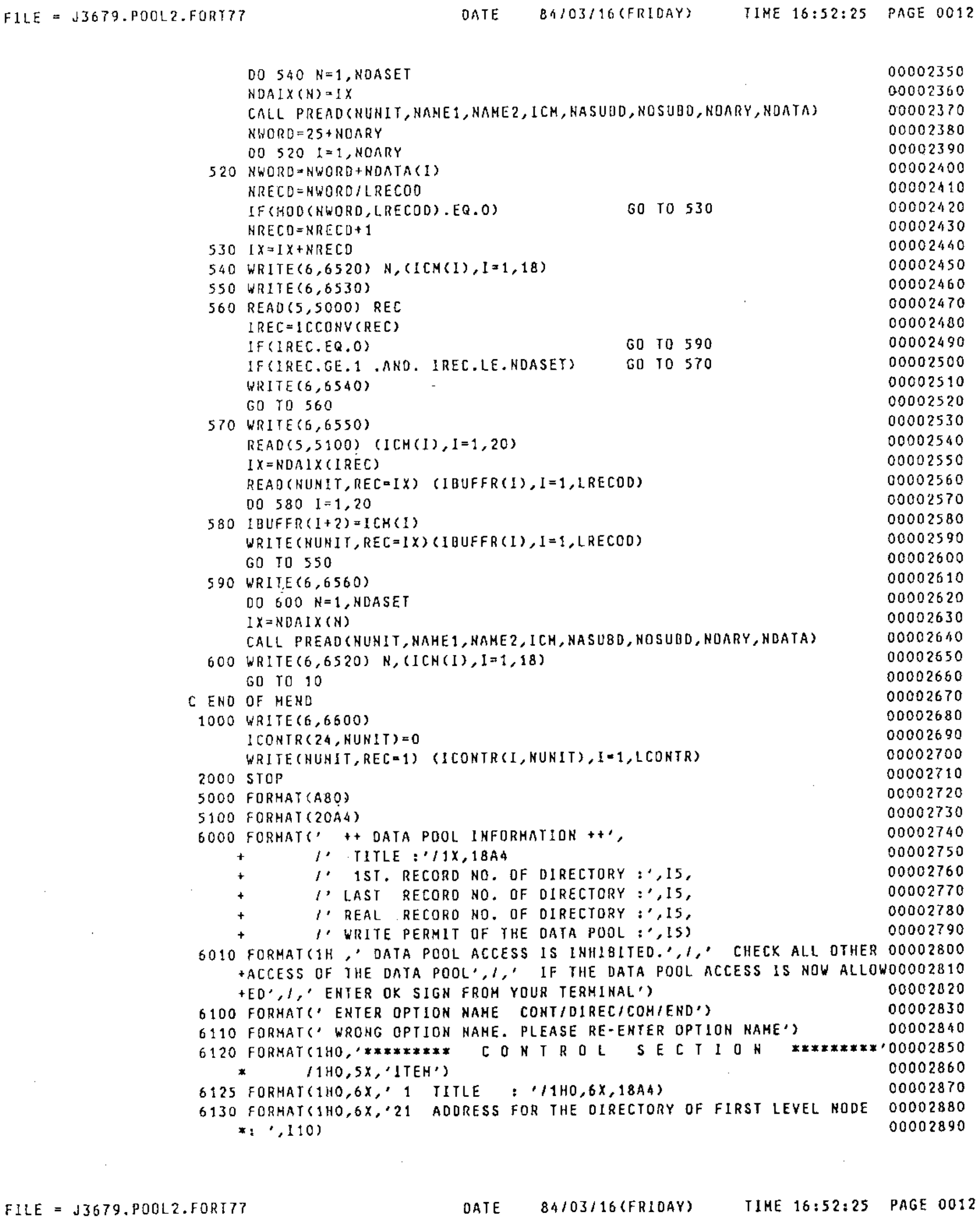
```
FILE = J3679.POOL2.FORT77                      DATE     84/03/16(FRIDAY)     TIME 16:52:25  PAGE 0006

                    C*                   INIT                                              *
                    C*                                                                     *
                    C***********************************************************************
                          PROGRAM     INIT
                          CHARACTER*80 IREC,TITLE
                          CHARACTER*8 NDATE,BLANK
                          DATA BLANK/'        '/
                       10 WRITE(6,300)
                          READ(5,100) IREC
                          NDIRCT=ICCONV(IREC)
                          IF(NDIRCT.EQ.-1)                GO TO 10
                          WRITE(6,400)
                          READ(5,200) (TITLE(I:I),I=1,64)
                          CALL DATE(NDATE)
                          TITLE(65:72)=BLANK
                          TITLE(73:80)=NDATE
                          LREC=900
                          CALL PINIT(91,NDIRCT,LREC,TITLE)
                          STOP
                      100 FORMAT(A80)
                      200 FORMAT(64A1)
                      300 FORMAT(' ENTER DIRECTORY SIZE ')
                      400 FORMAT(' ENTER TITLE (64 CHARACTERS)')
                          END

     **************
     ** LIST     **
     **************

                    C***********************************************************************
                    C*                                                                     *
                    C*                 LIST                                                *
                    C*                                                                     *
                    C***********************************************************************
                          PROGRAM   LIST
                          DIMENSION JCONTR(40),NDIRC(12),NODE(10),ICM(20),
                         *          NDATA(4)
                          COMMON /DPCONT/ LCONTR,NCONTR,ICONTR(40,99),IX,NSUBDS,NDSTAT
                    C
                          COMMON /DPWORK/ LBUFFR,LRECOD,IBUFFR(1000),NRECOD,NODE1,NODE2,
                         *                NADWN,NADAT,NDASET,NDATE(2),NINFON(5),NUTOLD,
                         *                NTHOLD,NODOLD(10,2),NA1
                          NUNIT=91
                          CALL POPEN(NUNIT,JCONTR)
                       10 WRITE(6,6000)
                       20 CALL NODINP(NTH,NODE)
                          IF(NTH.EQ.0)                    GO TO 1000
                          CALL PFIND(NUNIT,NODE,NTH,NDIRC,LLL)
                          IF(LLL.EQ.0)                    GO TO 100
                          WRITE(6,6010)
                          GO TO 20
                    C  PRINT DIRECTORY
                      100 WRITE(6,6020) (NODE(N),N=1,NTH)
                          CALL PRSUBD(NDIRC)
                    C  PRINT DATA

FILE = J3679.POOL2.FORT77                      DATE     84/03/16(FRIDAY)     TIME 16:52:25  PAGE 0006
```



```
FILE = J3679.POOL2.FORT77                  DATE   84/03/16(FRIDAY)     TIME 16:52:25  PAGE 0007

            NDASET=NDIRC(5)
            IF(NDASET .EQ. 0)              GO TO 200
            DO 150 N=1,NDASET
            CALL PREAD(NUNIT,NAME1,NAME2,ICH,NASBD,NOSBDS,NOARY,NDATA)
            WRITE(6,6040) N,(ICH(I),I=1,18),NOARY
            IF(NOARY.EQ.0) THEN
                 WRITE(6,6045)
            ELSE
                 WRITE(6,6050) (I,NDATA(I),I=1,NOARY)
            END IF
      C    SET ADDRESS FOR A NEXT DATA SET
            NWORD=25+NOARY
            DO 120 I=1,NOARY
        120 NWORD=NWORD + NDATA(I)
            NRECD=NWORD/LRECOD
            IF(MOD(NWORD,LRECOD).EQ.0)        GO TO 130
            NRECD=NRECD +1
        130 IX=IX + NRECD
        150 CONTINUE
        200 CONTINUE
            GO TO 10
      C
       1000 STOP
      C
       6000 FORMAT(' ENTER NODE NAME ')
       6010 FORMAT(' NOT FOUND THIS NODE NAME. PLEASE RE-ENTER NODE NAME')
       6020 FORMAT(' RECORD INFORMATIONS FOR NODE NAME ',A4,9('.',A4))
       6030 FORMAT('  ADDRESS OF A LOWER NODE DIRECTORY =',I6
          *       /'  ADDRESS OF A DATA SET            =',I6
          *       /'  LENGTH OF A DATA SET =',I4,'  NO. OF SUB-DATA SETS =',I4
          *       /'  DATA OF CREATION       =',2A4)
       6040 FORMAT(' ** INFORMATIONS FOR SUB-DATA SET ',I4,' **'/4X,18A4
          *       /'     NO. OF DATA ARRAIES =',I2)
       6045 FORMAT('     NOTHING OF DATA ARRAY')
       6050 FORMAT((4X,2('LENGTH OF DATA ',I1,'=',I6,3X)))
            END

**************
** MEND     **
**************

      C***********************************************************************
      C*                                                                      *
      C*                         MEND                                         *
      C*                                                                      *
      C***********************************************************************
            PROGRAM   MEND
            DIMENSION JCONTR(40)
            DIMENSION NODE(10),ICH(20),NDIRC(12),NDATA(4),NDAIX(200)
            COMMON /DPCONT/ LCONTR,NCONTR,ICONTR(40,99),IX,NSUBDS,NDSTAT
      C
            COMMON /DPWORK/ LBUFFR,LRECOD,IBUFFR(1000),NRECOD,NODE1,NODE2,
           *                NADWN,NADAT,NDASET,NDATE(2),NINFOM(5),NUTOLD,
           *                NTHOLD,NODOLD(10,2),NA1
            CHARACTER REC*80,CDAT*4,SLASH*4

FILE = J3679.POOL2.FORT77                  DATE   84/03/16(FRIDAY)     TIME 16:52:25  PAGE 0007
FILE = J3679.POOL2.FORT77                DATE     84/03/16(FRIDAY)     TIME 16:52:25  PAGE 0008

      DIMENSION RBUFFR(1)
      EQUIVALENCE (IBUFFR(1),RBUFFR(1))
      DATA SLASH/'/////'/
C
      NUNIT=91
      CALL POPEN(NUNIT,JCONTR)
      IDX=ICONTR(22,NUNIT)-1
      WRITE(6,6000) (ICONTR(J,NUNIT),J=1,18),ICONTR (21,NUNIT),IDX,
     +               ICONTR(30,NUNIT),ICONTR(24,NUNIT)
      IF(ICONTR(24,NUNIT).EQ.0)           GO TO 5
      WRITE(6,6010)
      READ( 5,5000,END=2000) REC
      IF(REC(1:2).EQ.'OK')                GO TO 5
      GO TO 2000
    5 ICONTR(24,NUNIT)=1
      WRITE(NUNIT,REC=1) (ICONTR(J,NUNIT),J=1,LCONTR)
   10 WRITE(6,6100)
   20 READ(5,5000,END=1000) REC
      IF(REC(1:4).EQ.'CONT')               GO TO 100
      IF(REC(1:5).EQ.'DIREC')              GO TO 200
      IF(REC(1:3).EQ.'COM')                GO TO 500
      IF(REC(1:3).EQ.'END')                GO TO 1000
      WRITE(6,6110)
      GO TO 20
C  MEND CONTROL SECTION
  100 N=0
      WRITE(6,6120)
      WRITE(6,6125) (ICONTR(LP1,NUNIT),LP1=1,18)
      WRITE(6,6130)  ICONTR(21,NUNIT)
      WRITE(6,6135)  ICONTR(22,NUNIT)
      WRITE(6,6140)  ICONTR(23,NUNIT)
      WRITE(6,6145)  ICONTR(24,NUNIT)
      WRITE(6,6150)  ICONTR(25,NUNIT)
      WRITE(6,6155)  ICONTR(26,NUNIT)
      WRITE(6,6160)  ICONTR(27,NUNIT)
      WRITE(6,6165)  ICONTR(28,NUNIT)
      WRITE(6,6170)  ICONTR(29,NUNIT)
      WRITE(6,6175)  ICONTR(30,NUNIT)
      WRITE(6,6180)  ICONTR(31,NUNIT)
  110 WRITE(6,6200)
  120 READ(5,5000) REC
      KC=ICCONV(REC)
      IF(KC.EQ.0)                          GO TO 150
      IF(KC.GE.1 .AND. KC.LE.31)           GO TO 130
      WRITE(6,6210)
      GO TO 120
  130 IF(KC.EQ.1) THEN
          WRITE(6,6215)
          READ(5,5100) (ICH(I),I=1,20)
          DO 140 I=1,20
  140     ICONTR(I,NUNIT)=ICH(I)
      ELSE
          WRITE(6,6220)
          READ(5,5000) REC
          IREC=ICCONV(REC)

FILE = J3679.POOL2.FORT77                DATE     84/03/16(FRIDAY)     TIME 16:52:25  PAGE 0008
FILE = J3679.POOL2.FORT77                  DATE    84/03/16(FRIDAY)      TIME 16:52:25  PAGE 0009

          IF(IREC.LT.0) THEN
              WRITE(6,6230)
              GO TO 120
          END IF
          ICONTR(KC,NUNIT)=IREC
      END IF
      N=N+1
      GO TO 110
  150 IF(N.EQ.0)                                GO TO 10
      WRITE(NUNIT,REC=1) (ICONTR(I,NUNIT),I=1,LCONTR)
      WRITE(6,6240)
      WRITE(6,6120)
      WRITE(6,6125) (ICONTR(LP1,NUNIT),LP1=1,18)
      WRITE(6,6130)  ICONTR(21,NUNIT)
      WRITE(6,6135)  ICONTR(22,NUNIT)
      WRITE(6,6140)  ICONTR(23,NUNIT)
      WRITE(6,6145)  ICONTR(24,NUNIT)
      WRITE(6,6150)  ICONTR(25,NUNIT)
      WRITE(6,6155)  ICONTR(26,NUNIT)
      WRITE(6,6160)  ICONTR(27,NUNIT)
      WRITE(6,6165)  ICONTR(28,NUNIT)
      WRITE(6,6170)  ICONTR(29,NUNIT)
      WRITE(6,6175)  ICONTR(30,NUNIT)
      WRITE(6,6180)  ICONTR(31,NUNIT)
      GO TO 10
C  MEND OF DIRECTORY SECTION
  200 WRITE(6,6300)
  210 CALL NODINP(NTH,NODE)
      IF(NTH.EQ.0)                              GO TO 10
      CALL PFIND(NUNIT,NODE,NTH,NDIRC,LLL)
      IF(LLL.EQ.0)                              GO TO 220
      WRITE(6,6310)
      GO TO 210
  220 WRITE(6,6320)
  230 READ(5,5000) REC
      IF(REC(1:3).EQ.'SUB')                     GO TO 250
      IF(REC(1:4).EQ.'HEAD')                    GO TO 410
      WRITE(6,6110)
      GO TO 230
C  MEND SUB-DIRECTORY
  250 NUPD=0
C     SEARCH SUB-DIRECTORY NO.
      ITEM=IBUFFR(4)
      DO 260 N=1,ITEM
          IF(IBUFFR(5+12*(N-1)).EQ.NODE(NTH)) THEN
          NSUB=N
          GO TO 270
          END IF
  260 CONTINUE
  270 CALL PRSUBD(NDIRC)
  275 WRITE(6,6330)
  280 READ(5,5000) REC
      IF(REC(1:3).EQ.'DEL')                     GO TO 290
      KC=ICCONV(REC)
      IF(KC.EQ.0)                               GO TO 400

FILE = J3679.POOL2.FORT77                  DATE    84/03/16(FRIDAY)      TIME 16:52:25  PAGE 0009
FILE = J3679.POOL2.FORT77                     DATE     84/03/16(FRIDAY)     TIME 16:52:25  PAGE 0010

            IF(KC.GE.1 .AND. KC.LE.12)                GO TO 320
            WRITE(6,6210)
            GO TO 280
      C     DELETE SUB-DIRECTORY
        290 IF(NSUB.EQ.ITEM)                          GO TO 310
            NN1=5+12*(NSUB-1)
            NN2=NN1+12
            DO 300 N=NSUB+1,ITEM
            DO 300 I=1,12
                IBUFFR(NN1)=IBUFFR(NN2)
                NN1=NN1+1
        300     NN2=NN2+1
        310 IBUFFR(4)=ITEM-1
            WRITE(NUNIT,REC=NA1) (IBUFFR(I),I=1,LRECOD)
            WRITE(6,6340) NODE(NTH)
            GO TO 10
      C     UPDATE SUB-DIRECTORY
        320 WRITE(6,6220)
            READ(5,5000) REC
            NUPD=NUPD+1
            NADD=4+12*(NSUB-1)
            IF(KC.EQ.1) THEN
                READ(REC,'(A4)') IBUFFR(NADD+KC)
                GO TO 275
                END IF
            IF(KC.GE.2 .AND. KC.LE.5) THEN
                IREC=ICCONV(REC)
                IBUFFR(NADD+KC)=IREC
                GO TO 275
                END IF
            IF(KC.EQ.6) THEN
                READ(REC,'(2A4)') IBUFFR(NADD+KC),IBUFFR(NADD+KC+1)
                GO TO 275
                END IF
            IF(KC.GE.8) THEN
                CALL CVDAT(REC,ITYP,IDAT,RDAT,CDAT)
                GO TO (330,340,350),ITYP
        330     IBUFFR(NADD+KC)=IDAT
                WRITE(6,6350) IDAT
                GO TO 275
        340     RBUFFR(NADD+KC)=RDAT
                WRITE(6,6360) RDAT
                GO TO 275
        350     READ(CDAT,'(A4)') IBUFFR(NADD+KC)
                WRITE(6,6370) CDAT
                GO TO 275
                END IF
        400 IF(NUPD.EQ.0)                             GO TO 10
            WRITE(NUNIT,REC=NA1) (IBUFFR(I),I=1,LRECOD)
            WRITE(6,6380)
            CALL PRSUBD(IBUFFR(NADD+1))
            GO TO 10
      C  MEND DIRECTORY HEAD
        410 NUPD=0
            IX=NDIRC(3)


FILE = J3679.POOL2.FORT77                     DATE     84/03/16(FRIDAY)     TIME 16:52:25  PAGE 0010
```



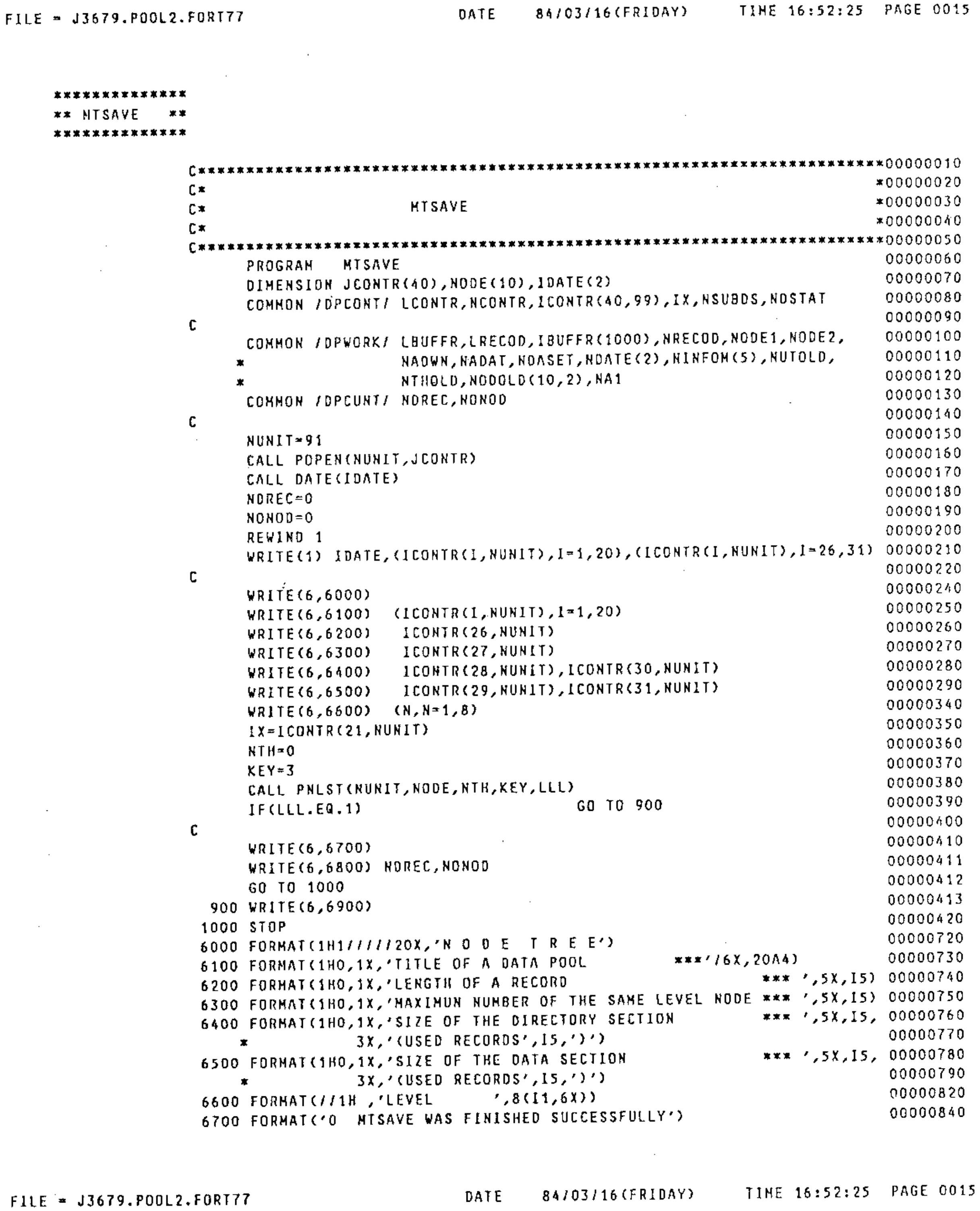
```
FILE = J3679.POOL2.FORT77                    DATE     84/03/16(FRIDAY)      TIME 16:52:25  PAGE 0011

      IF(IX.EQ.0) THEN
          WRITE(6,6400)
          GO TO 10
          END IF
      READ(NUNIT,REC=IX) (IBUFFR(I),I=1,LRECOD)
      WRITE(6,6410) (IBUFFR(I),I=1,4)
      ITEM=IBUFFR(4)
      WRITE(6,6420) (IBUFFR(5+12*(I-1)),I=1,ITEM)
  420 WRITE(6,6430)
  430 READ(5,5000) REC
      IF(REC(1:3).EQ.'DEL')                     GO TO 440
      KC=ICCONV(REC)
      IF(KC.EQ.0)                               GO TO 460
      IF(KC.GE.1 .AND. KC.LE.4)                 GO TO 450
      WRITE(6,6210)
      GO TO 430
C     DELETE DIRECTORY HEAD
  440 READ(SLASH,'(A4)') IBUFFR(1)
      IBUFFR(4)=0
      WRITE(NUNIT,REC=IX) (IBUFFR(I),I=1,LRECOD)
      WRITE(6,6440) NODE(NTH)
      GO TO 10
C     UPDATE DIRECTORY HEAD
  450 WRITE(6,6220)
      READ(5,5000) REC
      NUPD=NUPD+1
      IF(KC.LE.2) THEN
          READ(REC,'(A4)') IBUFFR(KC)
          GO TO 420
          END IF
      IF(KC.EQ.3 .OR. KC.EQ.4) THEN
          IREC=ICCONV(REC)
          IBUFFR(KC)=IREC
          GO TO 420
          END IF
  460 IF(NUPD.EQ.0)                             GO TO 10
      WRITE(NUNIT,REC=IX) (IBUFFR(I),I=1,LRECOD)
      WRITE(6,6450)
      WRITE(6,6410) (IBUFFR(I),I=1,4)
      GO TO 10
C  MEND OF COMMENT
  500 WRITE(6,6300)
  510 CALL NODINP(NTH,NODE)
      IF(NTH.EQ.0)                              GO TO 10
      CALL PFIND(NUNIT,NODE,NTH,NDIRC,LLL)
      IF(LLL.NE.0) THEN
          WRITE(6,6310)
          GO TO 510
          END IF
      NDASET=NDIRC(5)
      IF(NDASET.EQ.0) THEN
          WRITE(6,6500)
          GO TO 510
          END IF
      WRITE(6,6510) NDASET

FILE = J3679.POOL2.FORT77                    DATE     84/03/16(FRIDAY)      TIME 16:52:25  PAGE 0011
```



```
FILE = J3679.POOL2.FORT77                    DATE    84/03/16(FRIDAY)     TIME 16:52:25  PAGE 0012

      DO 540 N=1,NDASET
      NDAIX(N)=IX
      CALL PREAD(NUNIT,NAME1,NAME2,ICH,NASUBD,NOSUBD,NDARY,NDATA)
      NWORD=25+NDARY
      DO 520 I=1,NDARY
  520 NWORD=NWORD+NDATA(I)
      NRECD=NWORD/LRECOD
      IF(MOD(NWORD,LRECOD).EQ.0)             GO TO 530
      NRECD=NRECD+1
  530 IX=IX+NRECD
  540 WRITE(6,6520) N,(ICH(I),I=1,18)
  550 WRITE(6,6530)
  560 READ(5,5000) REC
      IREC=ICCONV(REC)
      IF(IREC.EQ.0)                          GO TO 590
      IF(IREC.GE.1 .AND. IREC.LE.NDASET)     GO TO 570
      WRITE(6,6540)
      GO TO 560
  570 WRITE(6,6550)
      READ(5,5100) (ICH(I),I=1,20)
      IX=NDAIX(IREC)
      READ(NUNIT,REC=IX) (IBUFFR(I),I=1,LRECOD)
      DO 580 I=1,20
  580 IBUFFR(I+2)=ICH(I)
      WRITE(NUNIT,REC=IX)(IBUFFR(I),I=1,LRECOD)
      GO TO 550
  590 WRITE(6,6560)
      DO 600 N=1,NDASET
      IX=NDAIX(N)
      CALL PREAD(NUNIT,NAME1,NAME2,ICH,NASUBD,NOSUBD,NDARY,NDATA)
  600 WRITE(6,6520) N,(ICH(I),I=1,18)
      GO TO 10
C END OF MEND
 1000 WRITE(6,6600)
      ICONTR(24,NUNIT)=0
      WRITE(NUNIT,REC=1) (ICONTR(I,NUNIT),I=1,LCONTR)
 2000 STOP
 5000 FORMAT(A80)
 5100 FORMAT(20A4)
 6000 FORMAT('  ++ DATA POOL INFORMATION ++',
     +        /'  TITLE :'/1X,18A4
     +        /'  1ST. RECORD NO. OF DIRECTORY :',I5,
     +        /' LAST  RECORD NO. OF DIRECTORY :',I5,
     +        /' REAL  RECORD NO. OF DIRECTORY :',I5,
     +        /' WRITE PERMIT OF THE DATA POOL :',I5)
 6010 FORMAT(1H ,' DATA POOL ACCESS IS INHIBITED.',/,'  CHECK ALL OTHER
     +ACCESS OF THE DATA POOL',/,'  IF THE DATA POOL ACCESS IS NOW ALLOW
     +ED',/,' ENTER OK SIGN FROM YOUR TERMINAL')
 6100 FORMAT(' ENTER OPTION NAME  CONT/DIREC/COM/END')
 6110 FORMAT(' WRONG OPTION NAME. PLEASE RE-ENTER OPTION NAME')
 6120 FORMAT(1H0,'*********    C O N T R O L    S E C T I O N    *********'
     *       /1H0,5X,'ITEM')
 6125 FORMAT(1H0,6X,' 1  TITLE    : '/1H0,6X,18A4)
 6130 FORMAT(1H0,6X,'21  ADDRESS FOR THE DIRECTORY OF FIRST LEVEL NODE
     *: ',I10)

FILE = J3679.POOL2.FORT77                    DATE    84/03/16(FRIDAY)     TIME 16:52:25  PAGE 0012
```

```
 6135 FORMAT(1H0,6X,'22  HEAD ADDRESS FOR THE VACANT DIRECTORY AREA      00002900
     *: ',I10)                                                          00002910
 6140 FORMAT(1H0,6X,'23  HEAD ADDRESS FOR THE VACANT DATA AREA           00002920
     *: ',I10)                                                          00002930
 6145 FORMAT(1H0,6X,'24  WRITE FLAG                                      00002940
     *: ',I10)                                                          00002950
 6150 FORMAT(1H0,6X,'25  READ FLAG (NOT USED)                            00002960
     *: ',I10)                                                          00002970
 6155 FORMAT(1H0,6X,'26  LENGTH OF THE ONE PHYSICAL RECORD               00002980
     *: ',I10)                                                          00002990
 6160 FORMAT(1H0,6X,'27  MAXIMUM NUMBER OF THE SAME LEVEL NODE           00003000
     *: ',I10)                                                          00003010
 6165 FORMAT(1H0,6X,'28  SIZE OF THE DIRECTORY SECTION                   00003020
     *: ',I10)                                                          00003030
 6170 FORMAT(1H0,6X,'29  SIZE OF THE DATA SECTION                        00003040
     *: ',I10)                                                          00003050
 6175 FORMAT(1H0,6X,'30  REAL NUMBER OF THE DIRECTORY RECORDS            00003060
     *: ',I10)                                                          00003070
 6180 FORMAT(1H0,6X,'31  REAL NUMBER OF THE DATA SET RECORDS             00003080
     *: ',I10)                                                          00003090
 6200 FORMAT(' ENTER ITEM NO. TO MEND. IF ENTER 0, END TO PROCESS')     00003100
 6210 FORMAT(' WRONG ITEM NO. PLEASE RE-ENTER ITEM NO.')                00003110
 6215 FORMAT(' ENTER NEW TITLE')                                        00003120
 6220 FORMAT(' ENTER NEW VALUE')                                        00003130
 6230 FORMAT(' WRONG VALUE WAS ENTERED. PLEASE RE-ENTER ITEM NO.')      00003140
 6240 FORMAT(' END OF MENDING A CONTROL SECTION SUCCESSFULLY')          00003150
 6300 FORMAT(' ENTER NODE NAME. IF ENTER NOTHING, END TO PROCESS')      00003160
 6310 FORMAT(' NOT FOUND NODE NAME. PLEASE RE-ENTER NODE NAME')         00003170
 6320 FORMAT(' ENTER OPTION NAME  SUB/HEAD')                            00003180
 6330 FORMAT(' ENTER ITEM NO. TO MEND OR DEL TO DELETE THIS SUB-DIRECTOR00003190
     *Y'/'  IF ENTER 0, END TO MEND THE SUB-DIRECTORY')                 00003200
 6340 FORMAT(' SUB-DIRECTORY ',A4,' WAS DELETED')                       00003210
 6350 FORMAT(' INPUT VALUE WAS INTEGER TYPE ',I5)                       00003220
 6360 FORMAT(' INPUT VALUE WAS REAL TYPE ',1PE12.3)                     00003230
 6370 FORMAT(' INPUT VALUE WAS CHARACTER TYPE ',A4)                     00003240
 6380 FORMAT(' END OF MENDING A SUB-DIRECTORY SECTION SUCCESSFULLY')    00003250
 6400 FORMAT(' DIRECTORY HEAD DID NOT EXIST.')                          00003260
 6410 FORMAT(' DIRECTORY HEAD'/                                         00003270
     1       ' ITEM     CONTENTS              '/                        00003280
     2       '   1   NODE NAME 1              ',1X,A4/                  00003290
     3       '   2   NODE NAME 2              ',1X,A4/                  00003300
     4       '   3   UPPER DIRECTORY ADDRESS ',I5/                      00003310
     5       '   4   NO. OF SUB-DIRECTORY    ',I5)                      00003320
 6420 FORMAT(' NODE NAMES FOR EACH SUB-DIRECTORY'/(12(2X,A4)))          00003330
 6430 FORMAT(' ENTER ITEM NO. TO MEND OR DEL TO DELETE THIS DIRECTORY HE00003340
     *AD'/   '  IF ENTER 0, END TO MEND THE DIRECTORY')                 00003350
 6440 FORMAT(' DIRECTORY HEAD',A4,' WAS DELETED')                       00003360
 6450 FORMAT(' END OF MENDING A DIRECTORY SECTION SUCCESSFULLY')        00003370
 6500 FORMAT(' THERE IS NO DATA RECORD. PLEASE RE-ENTER NODE NAME')     00003380
 6510 FORMAT(' NO. OF SUB-DATA-SET IS ',I3/                             00003390
     *       ' DAT NO.         COMMENT')                                00003400
 6520 FORMAT(I7,1X,18A4)                                                00003410
 6530 FORMAT(' ENTER DAT NO. TO MEND. IF ENTER 0, END TO PROCESS')      00003420
 6540 FORMAT(' WRONG DAT NO. PLEASE RE-ENTER DAT NO.')                  00003430
 6550 FORMAT(' ENTER NEW COMMENT')                                      00003440
```

```
FILE = J3679.POOL2.FORT77                DATE    84/03/16(FRIDAY)     TIME 16:52:25  PAGE 0014

 6560 FORMAT(' END OF MENDING DATA COMMENTS SUCCESSFULLY'/
     *        ' DAT NO.         COMMENT')
 6600 FORMAT(' END OF MEND COMMAND')
      END

**************
** MTCOPY   **
**************

C***********************************************************************
C*                                                                     *
C*                     MTCOPY                                          *
C*                                                                     *
C***********************************************************************
      PROGRAM    MTCOPY
      DIMENSION JCONTR(40),IDATE(2)
      COMMON /DPCONT/ LCONTR,NCONTR,ICONTR(40,99),IX,NSUBDS,NDSTAT
C
      COMMON /DPWORK/ LBUFFR,LRECOD,IBUFFR(1000),NRECOD,NODE1,NODE2,
     *                NADWN,NADAT,NDASET,NDATE(2),NINFOM(5),NUTOLD,
     *                NTHOLD,NODOLD(10,2),NA1
      CHARACTER*4 BLANK
      DATA BLANK/'    '/
C
      NUNIT=91
      CALL POPEN(NUNIT,JCONTR)
C RESET CONTROL SECTION
      REWIND 1
      READ(1) IDATE,(ICONTR(I,NUNIT),I=1,20)
      ICONTR(21,NUNIT)=2
      ICONTR(22,NUNIT)=3
      ICONTR(23,NUNIT)=ICONTR(28,NUNIT)+2
      ICONTR(24,NUNIT)=1
      ICONTR(30,NUNIT)=1
      ICONTR(31,NUNIT)=0
      NUTOLD=0
      NTHOLD=0
      NDSTAT=ICONTR(23,NUNIT)
      WRITE(NUNIT,REC=1) (ICONTR(LP1,NUNIT),LP1=1,LCONTR)
      READ(BLANK,'(A4)') IBUFFR(1)
      READ(BLANK,'(A4)') IBUFFR(2)
      IBUFFR(3)=0
      IBUFFR(4)=0
      WRITE(NUNIT,REC=2) (IBUFFR(LP1),LP1=1,4)
C LOAD BACK-UP DATA
      CALL PRECVR(NUNIT,2,LLL)
      WRITE(6,6000)
      CALL PWEND(NUNIT)
C
 1000 STOP
C
 6000 FORMAT('0 MTCOPY WAS FINISHED SUCCESSFULLY')
      END

FILE = J3679.POOL2.FORT77                DATE    84/03/16(FRIDAY)     TIME 16:52:25  PAGE 0014
```



```
FILE = J3679.POOL2.FORT77                    DATE    84/03/16(FRIDAY)     TIME 16:52:25  PAGE 0015


      **************
      ** MTSAVE   **
      **************

            C*********************************************************************00000010
            C*                                                                   *00000020
            C*                    MTSAVE                                         *00000030
            C*                                                                   *00000040
            C*********************************************************************00000050
                  PROGRAM   MTSAVE                                                00000060
                  DIMENSION JCONTR(40),NODE(10),IDATE(2)                          00000070
                  COMMON /DPCONT/ LCONTR,NCONTR,ICONTR(40,99),IX,NSUBDS,NDSTAT    00000080
            C                                                                     00000090
                  COMMON /DPWORK/ LBUFFR,LRECOD,IBUFFR(1000),NRECOD,NODE1,NODE2,  00000100
                 *                NADWN,NADAT,NDASET,NDATE(2),NINFOM(5),NUTOLD,   00000110
                 *                NTHOLD,NODOLD(10,2),NA1                         00000120
                  COMMON /DPCUNT/ NDREC,NONOD                                     00000130
            C                                                                     00000140
                  NUNIT=91                                                        00000150
                  CALL POPEN(NUNIT,JCONTR)                                        00000160
                  CALL DATE(IDATE)                                                00000170
                  NDREC=0                                                         00000180
                  NONOD=0                                                         00000190
                  REWIND 1                                                        00000200
                  WRITE(1) IDATE,(ICONTR(I,NUNIT),I=1,20),(ICONTR(I,NUNIT),I=26,31) 00000210
            C                                                                     00000220
                  WRITE(6,6000)                                                   00000240
                  WRITE(6,6100)  (ICONTR(I,NUNIT),I=1,20)                         00000250
                  WRITE(6,6200)   ICONTR(26,NUNIT)                                00000260
                  WRITE(6,6300)   ICONTR(27,NUNIT)                                00000270
                  WRITE(6,6400)   ICONTR(28,NUNIT),ICONTR(30,NUNIT)               00000280
                  WRITE(6,6500)   ICONTR(29,NUNIT),ICONTR(31,NUNIT)               00000290
                  WRITE(6,6600)  (N,N=1,8)                                        00000340
                  IX=ICONTR(21,NUNIT)                                             00000350
                  NTH=0                                                           00000360
                  KEY=3                                                           00000370
                  CALL PNLST(NUNIT,NODE,NTH,KEY,LLL)                              00000380
                  IF(LLL.EQ.1)                     GO TO 900                      00000390
            C                                                                     00000400
                  WRITE(6,6700)                                                   00000410
                  WRITE(6,6800) NDREC,NONOD                                       00000411
                  GO TO 1000                                                      00000412
              900 WRITE(6,6900)                                                   00000413
             1000 STOP                                                            00000420
             6000 FORMAT(1H1/////20X,'N O D E   T R E E')                         00000720
             6100 FORMAT(1H0,1X,'TITLE OF A DATA POOL           ***'/6X,20A4)     00000730
             6200 FORMAT(1H0,1X,'LENGTH OF A RECORD                   *** ',5X,I5) 00000740
             6300 FORMAT(1H0,1X,'MAXIMUN NUMBER OF THE SAME LEVEL NODE *** ',5X,I5) 00000750
             6400 FORMAT(1H0,1X,'SIZE OF THE DIRECTORY SECTION        *** ',5X,I5, 00000760
                 *          3X,'(USED RECORDS',I5,')')                            00000770
             6500 FORMAT(1H0,1X,'SIZE OF THE DATA SECTION             *** ',5X,I5, 00000780
                 *          3X,'(USED RECORDS',I5,')')                            00000790
             6600 FORMAT(//1H ,'LEVEL      ',8(I1,6X))                            00000820
             6700 FORMAT('0  MTSAVE WAS FINISHED SUCCESSFULLY')                   00000840


FILE = J3679.POOL2.FORT77                    DATE    84/03/16(FRIDAY)     TIME 16:52:25  PAGE 0015
```

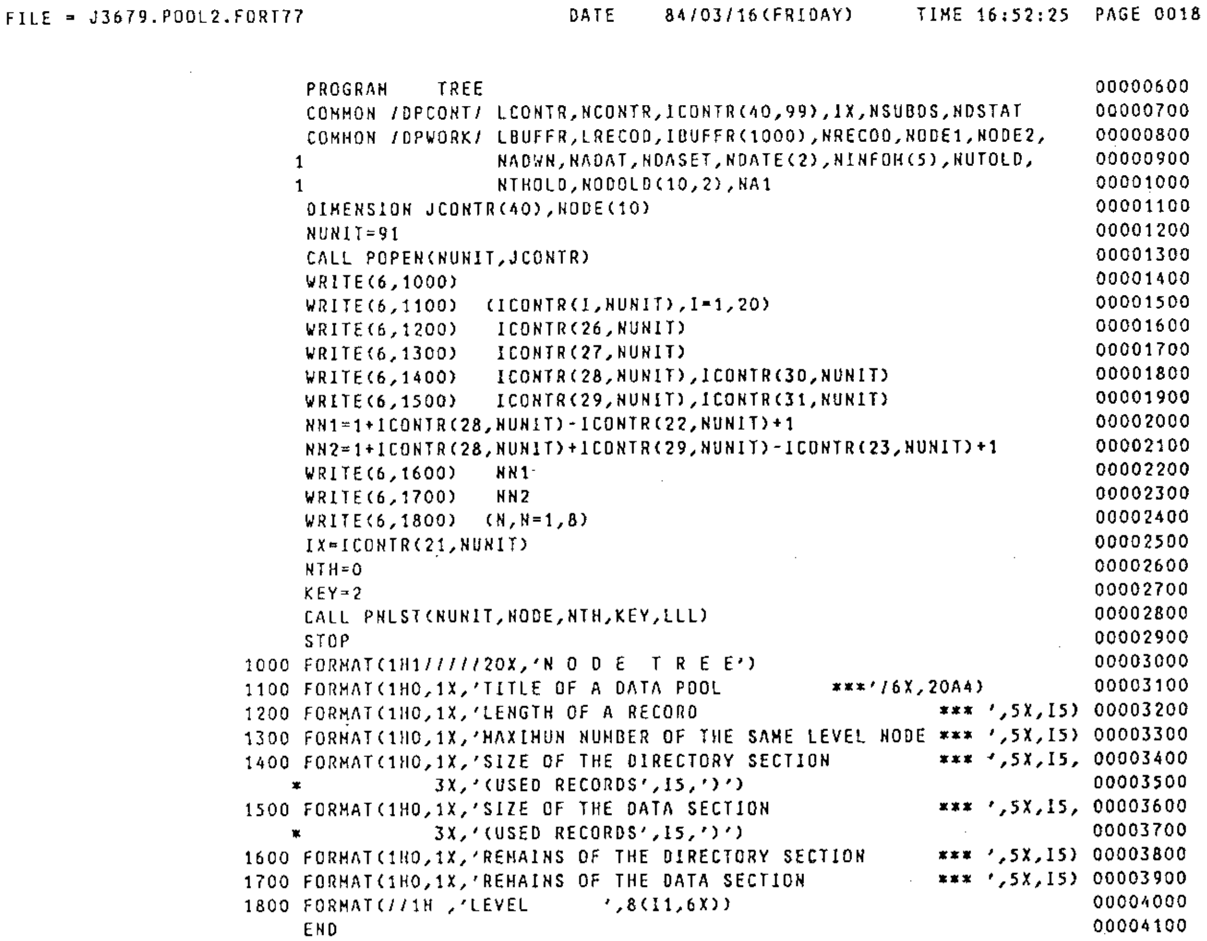
```
 6800 FORMAT('0    NO. OF DIRECTORY = ',I5/
     *       '    NO. OF NODE      = ',I5)
 6900 FORMAT('0  ABNORMAL END')
      END
```

```
**************
** RENAME   **
**************
```

```
C***********************************************************************
C*                                                                     *
C*                     RENAME                                          *
C*                                                                     *
C***********************************************************************
      PROGRAM   RENAME
      DIMENSION JCONTR(40),NODEA(10),NODEB(10),NDIRC(12)
      COMMON /DPCONT/ LCONTR,NCONTR,ICONTR(40,99),IX,NSUBDS,NDSTAT
C
      COMMON /DPWORK/ LBUFFR,LRECOD,IBUFFR(1000),NRECOD,NODE1,NODE2,
     *                NADOWN,NADAT,NDASET,NDATE(2),NINFOM(5),NUTOLD,
     *                NTHOLD,NODOLD(10,2),NA1
C
      NUNIT=91
      CALL POPEN(NUNIT,JCONTR)
   10 WRITE(6,6000)
      CALL NODINP(NTH1,NODEA)
      IF(NTH1.EQ.0)                     GO TO 1000
      WRITE(6,6010)
      CALL NODINP(NTH2,NODEB)
C  CHECK OF INPUT NODE
      IF(NTH1.EQ.NTH2)                  GO TO 100
      WRITE(6,6020)
      GO TO 10
  100 IF(NTH1.EQ.1)                     GO TO 120
      DO 110 N=1,NTH1-1
      IF(NODEA(N).EQ.NODEB(N))          GO TO 110
          WRITE(6,6030)
          GO TO 10
  110 CONTINUE
  120 IF(NODEA(NTH1).NE.NODEB(NTH1))    GO TO 130
      WRITE(6,6040)
      GO TO 10
  130 CALL PFIND(NUNIT,NODEB,NTH2,NDIRC,LLL)
      IF(LLL.NE.0)                      GO TO 140
      WRITE(6,6045)
      GO TO 10
  140 CALL PFIND(NUNIT,NODEA,NTH1,NDIRC,LLL)
      IF(LLL.EQ.0)                      GO TO 200
      WRITE(6,6050)
      GO TO 10
C  START RENAME
  200 ITEM=IBUFFR(4)
      DO 210 N=1,ITEM
      IF(IBUFFR(5+12*(N-1)).EQ.NODEA(NTH1)) THEN
          NSUB=N
          GO TO 220
          END IF
  210 CONTINUE
  220 IBUFFR(5+12*(NSUB-1))=NODEB(NTH2)
      NADWN=IBUFFR(7+12*(NSUB-1))
      NADAT=IBUFFR(8+12*(NSUB-1))
      NDASET=IBUFFR(9+12*(NSUB-1))
      WRITE(NUNIT,REC=NA1) (IBUFFR(I),I=1,LRECOD)
C   CHANGE LOWER NODE DIRECTORY
      IF(NADWN.EQ.0)                     GO TO 250
      IX=NADWN
      READ(NUNIT,REC=IX) (IBUFFR(I),I=1,LRECOD)
      IBUFFR(1)=NODEB(NTH2)
      WRITE(NUNIT,REC=IX) (IBUFFR(I),I=1,LRECOD)
C   CHANG DATA SECTION
  250 IF(NADAT.EQ.0)                     GO TO 300
      IX=NADAT
      DO 270 N=1,NDASET
      READ(NUNIT,REC=IX) (IBUFFR(I),I=1,LRECOD)
      IBUFFR(1)=NODEB(NTH2)
      NOARY=IBUFFR(25)
      WRITE(NUNIT,REC=IX)(IBUFFR(I),I=1,LRECOD)
      NWORD=25+NOARY
      DO 260 I=1,NOARY
  260 NWORD=NWORD+IBUFFR(25+I)
      NRECD=NWORD/LRECOD
      IF(MOD(NWORD,LRECOD).EQ.0)         GO TO 270
      NRECD=NRECD+1
  270 IX=IX+NRECD
  300 CONTINUE
      WRITE(6,6060)
      GO TO 10
C
 1000 STOP
C
 6000 FORMAT(' ENTER A OLD NODE NAME')
 6010 FORMAT(' ENTER A NEW NODE NAME')
 6020 FORMAT(' THE LEVEL OF TWO NODES ARE DIFFERENT. PLEASE RE-ENTER')
 6030 FORMAT(' A NODE NAME TO BE RENAMED IS ONLY LAST ONE. PLEASE RE-ENT
     *ER')
 6040 FORMAT(' A LAST NODE NAME IS SAME. PLEASE RE-ENTER')
 6045 FORMAT(' NEW NODE NAME EXIST ALREADY. PLEASE RE-ENTER')
 6050 FORMAT(' A OLD NODE NAME IS NOT FOUND. PLEASE RE-ENTER')
 6060 FORMAT(' RENAME IS FINISHED SUCCESSFULLY')
      END
```

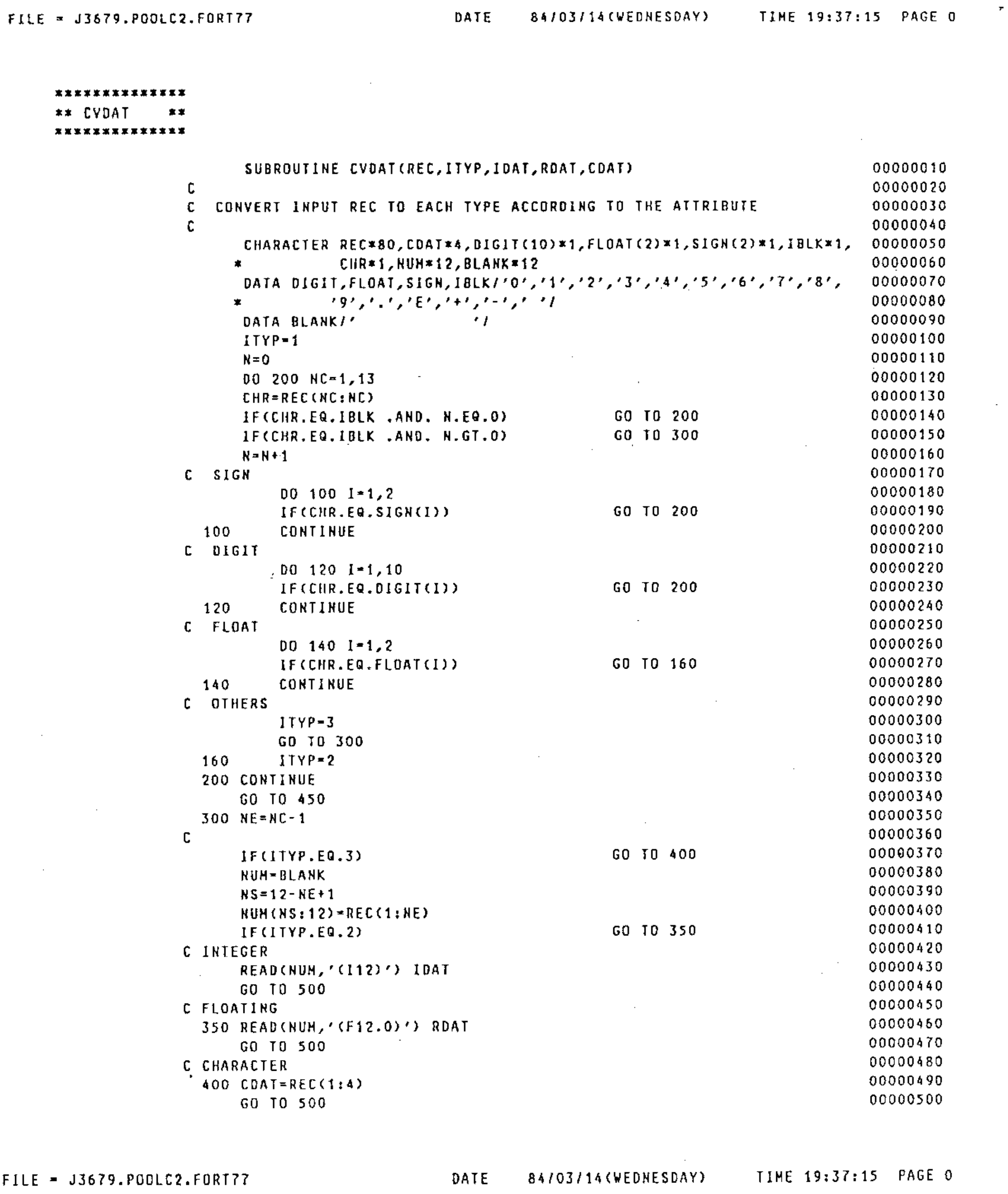
```
**************
** TREE     **
**************
```

```
C***********************************************************************
C*                                                                     *
C*                         TREE                                        *
C*                                                                     *
C***********************************************************************
```



```
FILE = J3679.POOL2.FORT77                 DATE     84/03/16(FRIDAY)     TIME 16:52:25  PAGE 0018

           PROGRAM    TREE
           COMMON /DPCONT/ LCONTR,NCONTR,ICONTR(40,99),IX,NSUBDS,NDSTAT
           COMMON /DPWORK/ LBUFFR,LRECOD,IBUFFR(1000),NRECOD,NODE1,NODE2,
          1                NADWN,NADAT,NDASET,NDATE(2),NINFOM(5),NUTOLD,
          1                NTHOLD,NODOLD(10,2),NA1
           DIMENSION JCONTR(40),NODE(10)
           NUNIT=91
           CALL POPEN(NUNIT,JCONTR)
           WRITE(6,1000)
           WRITE(6,1100)  (ICONTR(I,NUNIT),I=1,20)
           WRITE(6,1200)   ICONTR(26,NUNIT)
           WRITE(6,1300)   ICONTR(27,NUNIT)
           WRITE(6,1400)   ICONTR(28,NUNIT),ICONTR(30,NUNIT)
           WRITE(6,1500)   ICONTR(29,NUNIT),ICONTR(31,NUNIT)
           NN1=1+ICONTR(28,NUNIT)-ICONTR(22,NUNIT)+1
           NN2=1+ICONTR(28,NUNIT)+ICONTR(29,NUNIT)-ICONTR(23,NUNIT)+1
           WRITE(6,1600)   NN1
           WRITE(6,1700)   NN2
           WRITE(6,1800)  (N,N=1,8)
           IX=ICONTR(21,NUNIT)
           NTH=0
           KEY=2
           CALL PNLST(NUNIT,NODE,NTH,KEY,LLL)
           STOP
      1000 FORMAT(1H1/////20X,'N O D E  T R E E')
      1100 FORMAT(1H0,1X,'TITLE OF A DATA POOL          ***'/6X,20A4)
      1200 FORMAT(1H0,1X,'LENGTH OF A RECORD                         *** ',5X,I5)
      1300 FORMAT(1H0,1X,'MAXIMUM NUMBER OF THE SAME LEVEL NODE *** ',5X,I5)
      1400 FORMAT(1H0,1X,'SIZE OF THE DIRECTORY SECTION         *** ',5X,I5,
         *            3X,'(USED RECORDS',I5,')')
      1500 FORMAT(1H0,1X,'SIZE OF THE DATA SECTION              *** ',5X,I5,
         *            3X,'(USED RECORDS',I5,')')
      1600 FORMAT(1H0,1X,'REMAINS OF THE DIRECTORY SECTION      *** ',5X,I5)
      1700 FORMAT(1H0,1X,'REMAINS OF THE DATA SECTION           *** ',5X,I5)
      1800 FORMAT(//1H ,'LEVEL      ',8(I1,6X))
           END
```

```
                    DIRECTORY LIST OF J3679.POOLC2.FORT77

             ***************   ****************    ****************
               MEMBER NAME        PAGE NO.            NO. OF CARDS
             ***************   ****************    ****************

(NO.=001)    CVOAT                 0001                   57
(NO.=002)    ICCONV                0002                   26
(NO.=003)    NODINP                0002                   28
(NO.=004)    PNLST                 0003                   65
(NO.=005)    PRECVR                0004                   87
(NO.=006)    PRSUBD                0006                   56
(NO.=007)    PSAVE                 0007                   91
(NO.=008)    PWLIST                0009                   34
                                                   -----------------
                                                         444 CARDS
```



```
FILE = J3679.POOLC2.FORT77                  DATE    84/03/14(WEDNESDAY)     TIME 19:37:15  PAGE 0


      **************
      ** CVDAT    **
      **************

                          SUBROUTINE CVDAT(REC,ITYP,IDAT,RDAT,CDAT)
                    C
                    C  CONVERT INPUT REC TO EACH TYPE ACCORDING TO THE ATTRIBUTE
                    C
                          CHARACTER REC*80,CDAT*4,DIGIT(10)*1,FLOAT(2)*1,SIGN(2)*1,IBLK*1,
                         *          CHR*1,NUM*12,BLANK*12
                          DATA DIGIT,FLOAT,SIGN,IBLK/'0','1','2','3','4','5','6','7','8',
                         *         '9','.','E','+','-',' '/
                          DATA BLANK/'            '/
                          ITYP=1
                          N=0
                          DO 200 NC=1,13
                          CHR=REC(NC:NC)
                          IF(CHR.EQ.IBLK .AND. N.EQ.0)           GO TO 200
                          IF(CHR.EQ.IBLK .AND. N.GT.0)           GO TO 300
                          N=N+1
                    C  SIGN
                              DO 100 I=1,2
                              IF(CHR.EQ.SIGN(I))                 GO TO 200
                      100     CONTINUE
                    C  DIGIT
                              DO 120 I=1,10
                              IF(CHR.EQ.DIGIT(I))                GO TO 200
                      120     CONTINUE
                    C  FLOAT
                              DO 140 I=1,2
                              IF(CHR.EQ.FLOAT(I))                GO TO 160
                      140     CONTINUE
                    C  OTHERS
                              ITYP=3
                              GO TO 300
                      160     ITYP=2
                      200 CONTINUE
                          GO TO 450
                      300 NE=NC-1
                    C
                          IF(ITYP.EQ.3)                          GO TO 400
                          NUM=BLANK
                          NS=12-NE+1
                          NUM(NS:12)=REC(1:NE)
                          IF(ITYP.EQ.2)                          GO TO 350
                    C INTEGER
                          READ(NUM,'(I12)') IDAT
                          GO TO 500
                    C FLOATING
                      350 READ(NUM,'(F12.0)') RDAT
                          GO TO 500
                    C CHARACTER
                      400 CDAT=REC(1:4)
                          GO TO 500


FILE = J3679.POOLC2.FORT77                  DATE    84/03/14(WEDNESDAY)     TIME 19:37:15  PAGE 0
```

FILE = J3679.POOLC2.FORT77   DATE   84/03/14(WEDNESDAY)   TIME 19:37:15 PAGE 0

```
C ERROR
  450 WRITE(6,6000)
      IDAT=0
  500 RETURN
C
 6000 FORMAT(' INPUT DATA ERROR. DATA MUST BE IN COL. 1 - 12')
      END
```

```
**************
** ICCONV   **
**************
```

```
C-------------------------------------------------------------------C
C+++++++  F U N C T I O N   I C C O N V  +++++++++++++++++++++++++++C
C-------------------------------------------------------------------C
      FUNCTION ICCONV(N)
      CHARACTER*80 N
      CHARACTER*1  J(11)
      DATA J/'1','2','3','4','5','6','7','8','9','0',' '/
C
      MM=0
      DO 2 K=1,72
      DO 3 M=1,11
      IF(J(M).EQ.N(K:K)) GO TO 4
    3 CONTINUE
      GO TO 5
    4 CONTINUE
      IF(M.EQ.10) M=0
      IF(M.EQ.11) GO TO 2
      MM=MM*10+M
    2 CONTINUE
      ICCONV=MM
      RETURN
    5 WRITE(6,6)  N(K:K),K
      ICCONV=-1
      RETURN
    6 FORMAT(1H ,' +IRREGULAR CHARACTOR ',A1,' IN COL.',I3)
      END
```

```
**************
** NODINP   **
**************
```

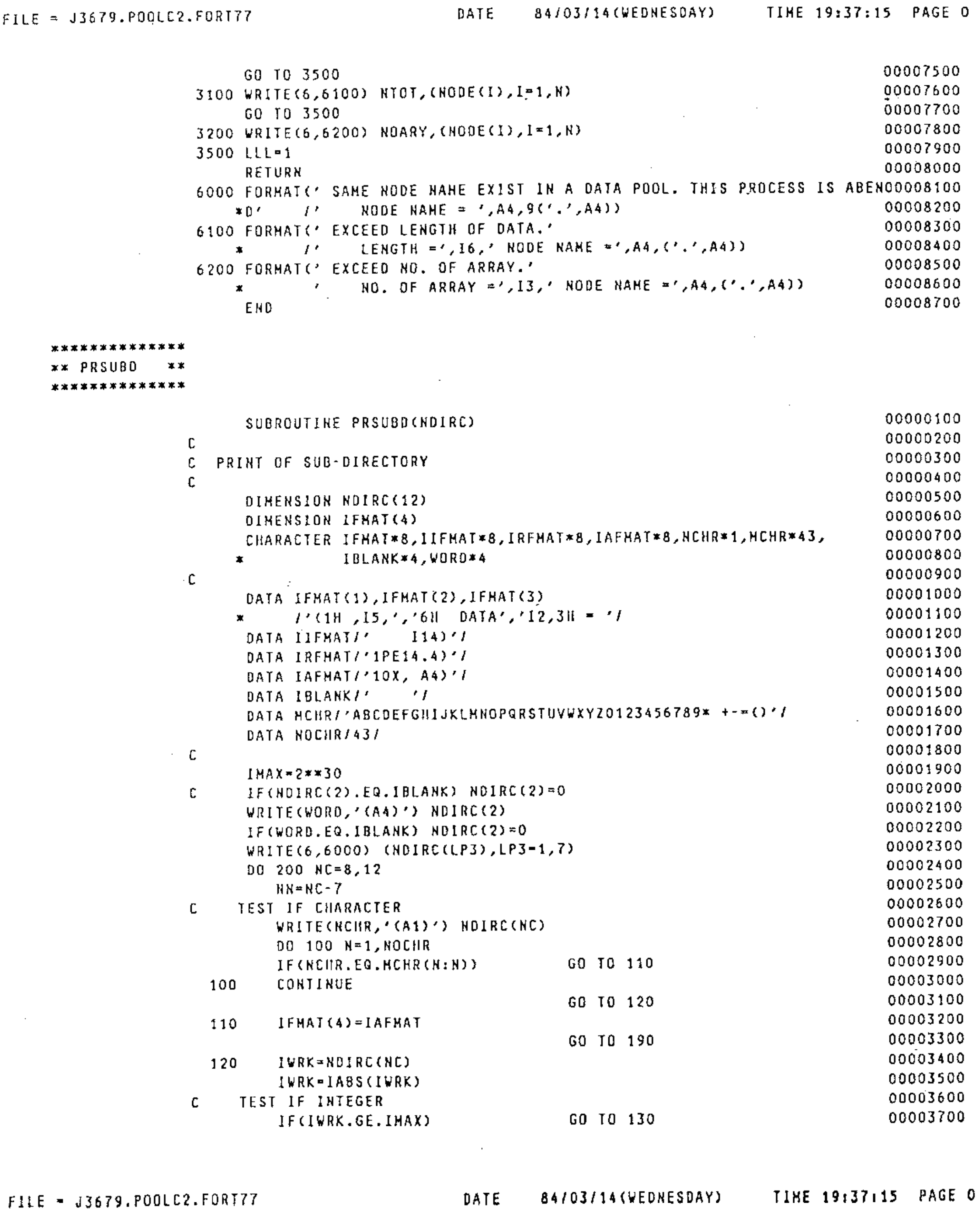
```
C
      SUBROUTINE NODINP(NTH,NODE)
C
C  READ NODE NAME LIST
C
      CHARACTER*4 NODE(10)
      CHARACTER   BUFF*80
C
   10 READ(5,6000,END=1000) BUFF
      N=0
      NTH=0
  100 N=N+1
```

FILE = J3679.POOLC2.FORT77   DATE   84/03/14(WEDNESDAY)   TIME 19:37:15 PAGE 0

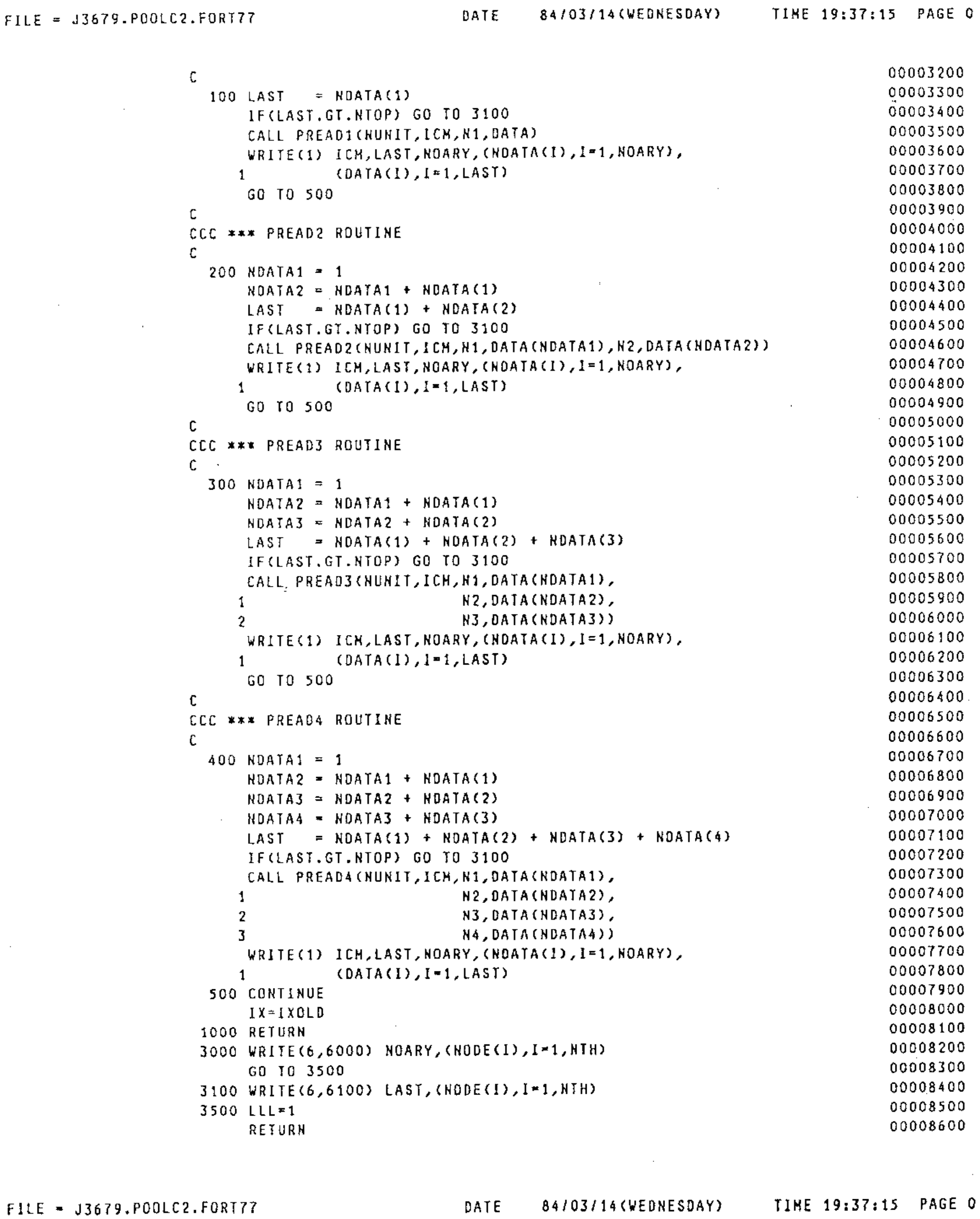
```
FILE = J3679.POOLC2.FORT77                    DATE      84/03/14(WEDNESDAY)      TIME 19:37:15  PAGE 0

               IF(BUFF(N:N+3).EQ.'    ')          GO TO 200                       00001300
               NTH=NTH+1                                                          00001400
               NODE(NTH)=BUFF(N:N+3)                                              00001500
               N=N+4                                                              00001600
               IF(BUFF(N:N).EQ.'.')               GO TO 100                       00001700
               IF(BUFF(N:N).EQ.' ')               GO TO 200                       00001800
               WRITE(6,6010)                                                      00001900
               GO TO 10                                                           00002000
           200 RETURN                                                             00002100
          1000 NTH=0                                                              00002200
               RETURN                                                             00002300
          6000 FORMAT(A80)                                                        00002400
          6010 FORMAT(' WRONG NODE NAME. INPUT NODE NAME MUST FOLLOW THE TYPE SHO00002500
              *WN BELOW'/         '   AAAA.BBBB.CCCC.DDDD   ...'/' PLEASE RE-ENTER00002600
              * NODE NAME')                                                       00002700
               END                                                                00002800

    **************
    ** PNLST    **
    **************

          C                                                                       00000100
                SUBROUTINE PNLST(NUNIT,NODE,NTH,KEY,LLL)                          00000200
          C                                                                       00000300
          C  SEARCH LOWER NODE NEST                                               00000400
          C         KEY=1  PUT DIRECTORY AND DATA TO A SEQUATIAL FILE (ITP1)      00000500
          C                AND NODE NAMES TO A FILE (ITP2)                        00000600
          C         KEY=2  PRINT OF NODE NAME TREE                                00000700
          C         KEY=3  BOTH                                                   00000800
          C                                                                       00000900
                DIMENSION NODE(10)                                                00001000
                COMMON /DPCONT/ LCONTR,NCONTR,ICONTR(40,99),IX,NSUBDS,NDSTAT      00001100
          C                                                                       00001200
                COMMON /DPWORK/ LBUFFR,LRECOD,IBUFFR(1000),NRECOD,NODE1,NODE2,    00001300
               *                NADWN,NADAT,NDASET,NDATE(2),NINFOM(5),NUTOLD,     00001400
               *                NTHOLD,NODOLD(10,2),NA1                           00001500
                COMMON /DPCUNT/ NDREC,NONOD                                       00001600
                DIMENSION ITEM(10),LL(10)                                         00001700
                LLL=0                                                             00001800
                NTHO=NTH                                                          00001900
                LVL=1                                                             00002000
                NTH=NTH+LVL                                                       00002100
                DO 10 N=1,10                                                      00002200
             10 LL(N)=1                                                           00002300
          C                                                                       00002400
            100 READ(NUNIT,REC=IX) (IBUFFR(I),I=1,LRECOD)                         00002500
                IF(LL(LVL).GT.1)                  GO TO 110                       00002600
                ITEM(LVL)=IBUFFR(4)                                               00002700
            110 IF(LL(LVL).GT.ITEM(LVL))          GO TO 250                       00002800
            120 NADD=4+12*(LL(LVL)-1)                                             00002900
                NODE(NTH)=IBUFFR(NADD+1)                                          00003000
                IX=IBUFFR(NADD+4)                                                 00003100
                IF(IBUFFR(NADD+5).EQ.0) IX=0                                      00003200
                IF(KEY.EQ.1 .OR. KEY.EQ.3) THEN                                   00003300
                    CALL PSAVE(NUNIT,NODE,NTH,IBUFFR(NADD+1),NRETUN)              00003400


FILE = J3679.POOLC2.FORT77                    DATE      84/03/14(WEDNESDAY)      TIME 19:37:15  PAGE 0
FILE = J3679.POOLC2.FORT77                    DATE    84/03/14(WEDNESDAY)      TIME 19:37:15  PAGE 0

            IF(NRETUN.NE.0)                GO TO 3100
         END IF
         IF(KEY.EQ.2 .OR. KEY.EQ.3) THEN
            CALL PWLIST(NUNIT,NODE,NTH)
         END IF
C       SET LOWER LEVEL NODE
         IX=IBUFFR(NADD+3)
         LL(LVL)=LL(LVL)+1
         IF(IX.EQ.0)                       GO TO 200
C       GO TO PROCESS OF LOWER NODE
         LVL=LVL+1
         NTH=NTH+1
         IF(LVL.GT.10)                     GO TO 3000
         NDREC=NDREC+1
         GO TO 100
C       PROCESS OF A NEXT SAME LEVEL NODE
    200 IF(LL(LVL).LE.ITEM(LVL))          GO TO 120
C       RETURN TO UPPER NODE
    250 LL(LVL)=1
         LVL=LVL-1
         NTH=NTH-1
         IF(LVL.EQ.0)                      GO TO 300
         IX=IBUFFR(3)
         GO TO 100
    300 NTH=NTHO
         RETURN
   3000 WRITE(6,6000)
   3100 LLL=1
         RETURN
   6000 FORMAT(' THE DEPTH OF NODE LEVEL IS TOO LARGE. MAX IS 10')
         END
```

```
**************
** PRECVR   **
**************
```

```
         SUBROUTINE PRECVR(IUNIT,KEY,LLL)
C
C       LOAD NODE DATA FROM BACK-UP FILE TO DATA POOL
C          KEY=1  IF SAME NODE NAME EXIST, ERROR RETURN
C              2  IF SAME NODE NAME EXIST, NOT OUTPUT NODE DATA AND
C                    CONTINUE THE PROCESS
         DIMENSION INFOH(5),ICH(20)
         DIMENSION NODE(10),NDATA(4)
         COMMON /PPP/ DATA(50000)
         COMMON /DPCONT/ LCONTR,NCONTR,ICONTR(40,99),IX,NSUBDS,NDSTAT
C
         COMMON /DPWORK/ LBUFFR,LRECOD,IBUFFR(1000),NRECOD,NODE1,NODE2,
        *                NADWN,NADAT,NDASET,NDATE(2),NINFOH(5),NUTOLD,
        *                NTHOLD,NODOLD(10,2),NA1
         COMMON /DPCUNT/ NDREC,NONOD
C
         LLL=0
     10 READ(1,END=700) N,(NODE(I),I=1,N),MADAT,MDASET,NDATE,INFOH
         CALL PSET(IUNIT,NODE,N,INFOH,0,NRETUN)

FILE = J3679.POOLC2.FORT77                    DATE    84/03/14(WEDNESDAY)      TIME 19:37:15  PAGE 0
FILE = J3679.POOLC2.FORT77                 DATE    84/03/14(WEDNESDAY)     TIME 19:37:15  PAGE 0

      IF(NRETUN.NE.0 .AND. KEY.EQ.1) GO TO 3000
      IF(NRETUN.NE.0 .AND. KEY.EQ.2) GO TO 550
      IF(HDASET.EQ.0) GO TO 600
      DO 500 I=1,HDASET
      READ(1) ICH,NTOT,NOARY,(NDATA(J),J=1,NOARY),
     1        (DATA(J),J=1,NTOT)
      IF(NTOT.GT.50000)           GO TO 3100
      NOARY1=NOARY+1
      GO TO ( 50,100,200,300,400),NOARY1
      GO TO 3200
   50 CALL PRITE(IUNIT,ICH)
      GO TO 500
C
CCC *** PRITE1 ROUTINE
C
  100 NDATA1 = 1
      CALL PRITE1(IUNIT,ICH,NDATA(1),DATA(NDATA1))
      GO TO 500
C
CCC *** PRITE2 ROUTINE
C
  200 NDATA1 = 1
      NDATA2 = NDATA1 + NDATA(1)
      CALL PRITE2(IUNIT,ICH,NDATA(1),DATA(NDATA1),NDATA(2),DATA(NDATA2))
      GO TO 500
C
CCC *** PRITE3 ROUTINE
C
  300 NDATA1 = 1
      NDATA2 = NDATA1 + NDATA(1)
      NDATA3 = NDATA2 + NDATA(2)
      CALL PRITE3(IUNIT,ICH,NDATA(1),DATA(NDATA1),NDATA(2),
     1           DATA(NDATA2),NDATA(3),DATA(NDATA3))
      GO TO 500
C
CCC *** PRITE4 ROUTINE
C
  400 NDATA1 = 1
      NDATA2 = NDATA1 + NDATA(1)
      NDATA3 = NDATA2 + NDATA(2)
      NDATA4 = NDATA3 + NDATA(3)
      CALL PRITE4(IUNIT,ICH,NDATA(1),DATA(NDATA1),
     1                      NDATA(2),DATA(NDATA2),
     1                      NDATA(3),DATA(NDATA3),
     1                      NDATA(4),DATA(NDATA4))
  500 CONTINUE
      GO TO 600
  550 IF(HDASET.EQ.0)                 GO TO 600
      DO 560 I=1,HDASET
  560 READ(1)
  600 CONTINUE
      GO TO 10
  700 CONTINUE
      RETURN
 3000 WRITE(6,6000) (NODE(I),I=1,N)


FILE = J3679.POOLC2.FORT77                 DATE    84/03/14(WEDNESDAY)     TIME 19:37:15  PAGE 0
```



```
FILE = J3679.POOLC2.FORT77                 DATE    84/03/14(WEDNESDAY)     TIME 19:37:15  PAGE 0

      GO TO 3500
3100  WRITE(6,6100) NTOT,(NODE(I),I=1,N)
      GO TO 3500
3200  WRITE(6,6200) NOARY,(NODE(I),I=1,N)
3500  LLL=1
      RETURN
6000  FORMAT(' SAME NODE NAME EXIST IN A DATA POOL. THIS PROCESS IS ABEN
     *D'    /'     NODE NAME = ',A4,9('.',A4))
6100  FORMAT(' EXCEED LENGTH OF DATA.'
     *       /'    LENGTH =',I6,' NODE NAME =',A4,('.',A4))
6200  FORMAT(' EXCEED NO. OF ARRAY.'
     *       '     NO. OF ARRAY =',I3,' NODE NAME =',A4,('.',A4))
      END

**************
** PRSUBD   **
**************

      SUBROUTINE PRSUBD(NDIRC)
C
C  PRINT OF SUB-DIRECTORY
C
      DIMENSION NDIRC(12)
      DIMENSION IFMAT(4)
      CHARACTER IFMAT*8,IIFMAT*8,IRFMAT*8,IAFMAT*8,NCHR*1,MCHR*43,
     *          IBLANK*4,WORD*4
C
      DATA IFMAT(1),IFMAT(2),IFMAT(3)
     *     /'(1H ,I5,','6H  DATA','I2,3H = '/
      DATA IIFMAT/'    I14)'/
      DATA IRFMAT/'1PE14.4)'/
      DATA IAFMAT/'10X, A4)'/
      DATA IBLANK/'    '/
      DATA MCHR/'ABCDEFGHIJKLMNOPQRSTUVWXYZ0123456789* +-=()'/
      DATA NOCHR/43/
C
      IMAX=2**30
C     IF(NDIRC(2).EQ.IBLANK) NDIRC(2)=0
      WRITE(WORD,'(A4)') NDIRC(2)
      IF(WORD.EQ.IBLANK) NDIRC(2)=0
      WRITE(6,6000) (NDIRC(LP3),LP3=1,7)
      DO 200 NC=8,12
         NN=NC-7
C    TEST IF CHARACTER
         WRITE(NCHR,'(A1)') NDIRC(NC)
         DO 100 N=1,NOCHR
         IF(NCHR.EQ.MCHR(N:N))          GO TO 110
  100    CONTINUE
                                        GO TO 120
  110    IFMAT(4)=IAFMAT
                                        GO TO 190
  120    IWRK=NDIRC(NC)
         IWRK=IABS(IWRK)
C    TEST IF INTEGER
         IF(IWRK.GE.IMAX)               GO TO 130

FILE = J3679.POOLC2.FORT77                 DATE    84/03/14(WEDNESDAY)     TIME 19:37:15  PAGE 0
FILE = J3679.POOLC2.FORT77                 DATE    84/03/14(WEDNESDAY)     TIME 19:37:15  PAGE 0

             IFMAT(4)=IIFMAT
                                             GO TO 190
   130       IFMAT(4)=IRFMAT
   190       CONTINUE
             WRITE(6,IFMAT) NC,NN,NDIRC(NC)
   200     CONTINUE
C
C----       FORMAT
C
  6000 FORMAT('  ITEM          CONTENTS'
      *         /'      1  NODE NAME                 =',4X,A4
      *         /'      2  TOTAL LENG. OF DATA SET =',I8
      *         /'      3  ADDRESS OF A LOWER NODE =',I8
      *         /'      4  ADDRESS OF A DATA SET   =',I8
      *         /'      5  NO. OF SUB-DATA SETS    =',I8
      *         /'      6  DATE OF CREATION        =',2A4)
C
       RETURN
       END

**************
** PSAVE    **
**************

       SUBROUTINE PSAVE(NUNIT,NODE,NTH,NDIRC,LLL)
C
C    SAVE NODE DATA TO A SEQUANTIAL FILE
C
       DIMENSION NODE(1),NDIRC(1)
       COMMON /DPCONT/ LCONTR,NCONTR,ICONTR(40,99),IX,NSUBDS,NDSTAT
       COMMON /DPCUNT/ NDREC,NONOD
       COMMON /PPP/ DATA(50000)
       DIMENSION NDATA(4),ICM(20)
C
       LLL=0
       IXOLD=IX
       NTOP = 50000
       WRITE(1) NTH,(NODE(I),I=1,NTH),(NDIRC(I),I=4,12)
       WRITE(2) NTH,(NODE(I),I=1,NTH)
       NONOD=NONOD+1
       IF(NDIRC(5).EQ.0)                GO TO 1000
       DO 500 N=1,NDIRC(5)
       CALL PREAD(NUNIT,NAME1,NAME2,ICM,NASDD,NOSDDS,NOARY,NDATA)
C
       NOARY1=NOARY+1
       GO TO (50,100,200,300,400),NOARY1
       GO TO 3000
C
    50 NDATA(1) = 0
       DATA(1)=0.0
       WRITE(1) ICM,NDATA(1),NOARY,(NDATA(I),I=1,NOARY),
     1          (DATA(I),I=1,NDATA(1))
       GO TO 500
C
CCC *** PREAD1 ROUTINE

FILE = J3679.POOLC2.FORT77                 DATE    84/03/14(WEDNESDAY)     TIME 19:37:15  PAGE 0
```



FILE = J3679.POOLC2.FORT77   DATE   84/03/14(WEDNESDAY)   TIME 19:37:15 PAGE 0

```
C
 100  LAST   = NDATA(1)
      IF(LAST.GT.NTOP) GO TO 3100
      CALL PREAD1(NUNIT,ICH,N1,DATA)
      WRITE(1) ICH,LAST,NOARY,(NDATA(I),I=1,NOARY),
     1         (DATA(I),I=1,LAST)
      GO TO 500
C
CCC *** PREAD2 ROUTINE
C
 200  NDATA1 = 1
      NDATA2 = NDATA1 + NDATA(1)
      LAST   = NDATA(1) + NDATA(2)
      IF(LAST.GT.NTOP) GO TO 3100
      CALL PREAD2(NUNIT,ICH,N1,DATA(NDATA1),N2,DATA(NDATA2))
      WRITE(1) ICH,LAST,NOARY,(NDATA(I),I=1,NOARY),
     1         (DATA(I),I=1,LAST)
      GO TO 500
C
CCC *** PREAD3 ROUTINE
C
 300  NDATA1 = 1
      NDATA2 = NDATA1 + NDATA(1)
      NDATA3 = NDATA2 + NDATA(2)
      LAST   = NDATA(1) + NDATA(2) + NDATA(3)
      IF(LAST.GT.NTOP) GO TO 3100
      CALL PREAD3(NUNIT,ICH,N1,DATA(NDATA1),
     1                      N2,DATA(NDATA2),
     2                      N3,DATA(NDATA3))
      WRITE(1) ICH,LAST,NOARY,(NDATA(I),I=1,NOARY),
     1         (DATA(I),I=1,LAST)
      GO TO 500
C
CCC *** PREAD4 ROUTINE
C
 400  NDATA1 = 1
      NDATA2 = NDATA1 + NDATA(1)
      NDATA3 = NDATA2 + NDATA(2)
      NDATA4 = NDATA3 + NDATA(3)
      LAST   = NDATA(1) + NDATA(2) + NDATA(3) + NDATA(4)
      IF(LAST.GT.NTOP) GO TO 3100
      CALL PREAD4(NUNIT,ICH,N1,DATA(NDATA1),
     1                      N2,DATA(NDATA2),
     2                      N3,DATA(NDATA3),
     3                      N4,DATA(NDATA4))
      WRITE(1) ICH,LAST,NOARY,(NDATA(I),I=1,NOARY),
     1         (DATA(I),I=1,LAST)
 500  CONTINUE
      IX=IXOLD
 1000 RETURN
 3000 WRITE(6,6000) NOARY,(NODE(I),I=1,NTH)
      GO TO 3500
 3100 WRITE(6,6100) LAST,(NODE(I),I=1,NTH)
 3500 LLL=1
      RETURN
```

FILE = J3679.POOLC2.FORT77   DATE   84/03/14(WEDNESDAY)   TIME 19:37:15 PAGE 0

```
FILE = J3679.POOLC2.FORT77                    DATE     84/03/14(WEDNESDAY)      TIME 19:37:15  PAGE 0

                 6000 FORMAT(' EXCEED NO. OF ARRAY.'
                     *      /'   NO.OF ARRAY =',I3,' NODE NAME =',A4,9('.',A4))
                 6100 FORMAT(' EXCEED LENGTH OF DATA.'
                     *      /'   LENGTH =',I6,' NODE NAME =',A4,9('.',A4))
                          END

     **************
     ** PWLIST   **
     **************

                          SUBROUTINE PWLIST(NUNIT,NODE,NO)
                    C
                    C  PRINT OF NODE TREE
                    C
                          COMMON /DPCONT/ LCONTR,NCONTR,ICONTR(40,99),IX,NSUBDS,NDSTAT
                          DIMENSION NODE(10),NDATA(4)
                          CHARACTER*4 ICH(20),BLANK
                          DATA BLANK/'    '/
                    C
                          IF(NO.GT.8)                        GO TO 1000
                          IF(IX.EQ.0) THEN
                               DO 10 N=1,20
                       10      ICH(N)=BLANK
                          ELSE
                               CALL PREAD(NUNIT,NAME1,NAME2,ICH,NASBD,NOSBDS,NOARY,NDATA)
                          ENDIF
                          IF(NO.EQ.1) WRITE(6,1100) NODE(NO),(ICH(I),I=1,18)
                          IF(NO.EQ.2) WRITE(6,1200) NODE(NO),(ICH(I),I=1,18)
                          IF(NO.EQ.3) WRITE(6,1300) NODE(NO),(ICH(I),I=1,18)
                          IF(NO.EQ.4) WRITE(6,1400) NODE(NO),(ICH(I),I=1,18)
                          IF(NO.EQ.5) WRITE(6,1500) NODE(NO),(ICH(I),I=1,18)
                          IF(NO.EQ.6) WRITE(6,1600) NODE(NO),(ICH(I),I=1,18)
                          IF(NO.EQ.7) WRITE(6,1700) NODE(NO),(ICH(I),I=1,18)
                          IF(NO.EQ.8) WRITE(6,1800) NODE(NO),(ICH(I),I=1,16)
                 1000 RETURN
                 1100 FORMAT(    1H / 7X,           A4,1X,':',18A4)
                 1200 FORMAT( 8X,'I'/ 8X,'I-----',A4,1X,':',18A4)
                 1300 FORMAT(15X,'I'/15X,'I-----',A4,1X,':',18A4)
                 1400 FORMAT(22X,'I'/22X,'I-----',A4,1X,':',18A4)
                 1500 FORMAT(29X,'I'/29X,'I-----',A4,1X,':',18A4)
                 1600 FORMAT(36X,'I'/36X,'I-----',A4,1X,':',18A4)
                 1700 FORMAT(43X,'I'/43X,'I-----',A4,1X,':',18A4)
                 1800 FORMAT(50X,'I'/50X,'I-----',A4,1X,':',16A4)
                          END
```

```
FILE = J3679.TSSMAC.CNTL                    DATE   84/03/16(FRIDAY)     TIME 17:00:35  PAGE 0001


**************
** COND     **
**************

              //JCLG  JOB                                                               00000010
              //*******************************************************************    00000020
              //*         JOB CONTROL LANGAGE FOR CONDENSE COMMAND                  *   00000030
              //*                                                                   *   00000040
              //*     PLEASE CHANGE JUSER CARD, PASSWORD, DATA POOL NAME AND        *   00000050
              //*         BACK-UP FILE NAME                                         *   00000060
              //*     AT END OF CHANGE PLEASE ENTER SUBMIT COMMAND                  *   00000061
              //*                                                                   *   00000070
              //*******************************************************************    00000080
              // EXEC JCLG                                                              00000090
              //SYSIN DD DATA,DLM='++'                                                  00000100
              // JUSER ????????,XX.XXXXXX,YYYY.ZZZ                                      00000110
               T.4 C.1 W.0 I.5 P.0 OPN                                                  00000120
               OPTP PASSWORD=??                                                         00000130
              // EXEC LMGO,LM='J3679.POOLX',PNM=COND                                    00000140
              //*    DATA POOL                                                          00000150
              //*     CHANGE DSN:DATA SET NAME                                          00000160
              // EXPAND DISKTO,DDN=FT91F001,DSN='JXXXX.????????',MODE=OUT               00000170
              //*    BACK-UP FILE                                                       00000180
              //*     CHANGE DSN:DATA SET NAME                                          00000190
              // EXPAND DISKTN,DDN=FT01F001,DSN='JXXXX.@@BACKUP',UNIT=TSSWK,            00000220
              //            SPC='500,300'                                               00000230
              // EXPAND DISK,DDN=FT02F001                                               00000240
              ++                                                                        00000250
              //                                                                        00000260

**************
** HTCOPY   **
**************

              //JCLG  JOB                                                               00000010
              //*******************************************************************    00000020
              //*         JOB CONTROL LANGAGE FOR HTCOPY COMMAND                    *   00000030
              //*                                                                   *   00000040
              //*     PLEASE CHANGE JUSER CARD, PASSWORD, DATA POOL NAME AND        *   00000050
              //*         BACK-UP TAPE NAME                                         *   00000060
              //*     AT END OF CHANGE PLEASE ENTER SUBMIT COMMAND                  *   00000061
              //*                                                                   *   00000070
              //*******************************************************************    00000080
              // EXEC JCLG                                                              00000090
              //SYSIN DD DATA,DLM='++'                                                  00000100
              // JUSER ????????,XX.XXXXXX,YYYY.ZZZ                                      00000110
               T.4 C.1 W.0 I.5 P.0 OPN MTU                                              00000120
               OPTP PASSWORD=??                                                         00000130
              // EXEC LMGO,LM='J3679.POOLX',PNM=HTCOPY                                  00000140
              //*    DATA POOL                                                          00000150
              //*     CHANGE DSN:DATA SET NAME                                          00000160
              // EXPAND DISKTO,DDN=FT91F001,DSN='JXXXX.????????',MODE=OUT               00000170
              //*    BACK-UP TAPE                                                       00000180
              //*     CHANGE DSN:DATA SET NAME                                          00000190


FILE = J3679.TSSMAC.CNTL                    DATE   84/03/16(FRIDAY)     TIME 17:00:35  PAGE 0001
FILE = J3679.TSSMAC.CNTL                       DATE    84/03/16(FRIDAY)     TIME 17:00:35  PAGE 0002

               //*          MTV:VOLUME NUMBER OF A TAPE                               00000200
               //*          POS:DATA SET POSITION ON A TAPE                           00000210
               // EXPAND TAPE,DDN=FT01F001,DSN='JXXXX.????????',MTV=??????,MTU=TAPE,   00000220
               //        POS=?                                                        00000230
               ++                                                                     00000240
               //                                                                     00000250

    **************
    ** MTSAVE   **
    **************

               //JCLG  JOB                                                            00000010
               //*****************************************************************     00000020
               //*        JOB CONTROL LANGAGE FOR MTSAVE COMMAND                 *     00000030
               //*                                                               *     00000040
               //*    PLEASE CHANGE JUSER CARD, PASSWORD, DATA POOL NAME AND     *     00000050
               //*        BACK-UP TAPE NAME.                                     *     00000060
               //*    AT END OF CHANGE PLEASE ENTER SUBMIT COMMAND               *     00000061
               //*                                                               *     00000070
               //*****************************************************************     00000080
               // EXEC JCLG                                                           00000090
               //SYSIN DD DATA,DLM='++'                                               00000100
               // JUSER ????????,XX.XXXXXX,YYYY.ZZZ                                   00000110
                T.4 C.1 W.0 I.5 P.0 OPN MTU                                           00000120
                OPTP PASSWORD=??                                                      00000130
               // EXEC LMGO,LM='J3679.POOLX',PNM=MTSAVE                               00000140
               //*   DATA POOL                                                        00000150
               //*    CHANGE DSN:DATA SET NAME                                        00000160
               // EXPAND DISKTO,DDN=FT91F001,DSN='JXXXX.????????'                     00000170
               //*   BACK-UP TAPE                                                     00000180
               //*    CHANGE DSN:DATA SET NAME                                        00000190
               //*          MTV:VOLUME NUMBER OF A TAPE                               00000200
               //*          POS:DATA SET POSITION ON A TAPE                           00000210
               // EXPAND TAPE,DDN=FT01F001,DSN='JXXXX.????????',MTV=??????,MTU=TAPE,   00000220
               //        POS=?,DISP='NEW,PASS'                                        00000230
               // EXPAND DISK,DDN=FT02F001                                            00000240
               ++                                                                     00000250
               //                                                                     00000260
```