JAERI-M
86-107

USER'S MANUAL OF THE MKENO-DAR CODE SYSTEM

August 1986

Yoshitaka NAITO and Hirochika NAKAE

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.

編集兼発行　日本原子力研究所
印　　刷　いばらき印刷㈱

JAERI-M 86-107

USER'S MANUAL OF THE MKENO-DAR CODE SYSTEM

Yoshitaka NAITO and Hirochika NAKAE

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

(Received July 7, 1986)

The computer code manual of MKENO-DAR which is a direct angular representation monte carlo code for criticality safety analysis is already issued as JAERI-M report, however, complex pre-stage data handlings and calculations by auxiliary programs are required before the execution of MKENO-DAR.

The MKENO-DAR CODE SYSTEM widely spans a whole code system including MKENO-DAR and other pre-stage auxiliary programs.

This report discusses the systematic treatment of the MKENO-DAR CODE SYSTEM and shows the simplified calculation technique from the user side of view.

Keywords: MKENO-DAR, Computer, Code, Criticality Safety Analysis, Monte Calro, Manual

JAERI－M 86 － 107

MKENO－DARコードシステムの使用手引

日本原子力研究所東海研究所燃料安全工学部
内藤　俶孝・中江　廣親

（1986年7月7日受理）

臨界安全解析のための直接角度表示モンテカルロコードMKENO－DARの使用手引はJAERI－Mレポートにより，すでに紹介されている。しかしながら，MKENO－DARの実行については，それ以前に補助プログラムによる煩雑なデータの取り扱いと計算が要求される。

MKENO－DARコードシステムは，MKENO－DARと，その補助プログラムを含む全体系を示すものである。

本報告は，MKENO－DARコードシステムの体系的な取り扱いと，単純化した計算手法を，利用者の立場から説明したものである。

東海研究所　〒319－11 茨城県那珂郡東海村白方字白根2－4

CONTENTS

目　　次

1. INTRODUCTION

Improving the Monte-Carlo code MULTI-KENO, the MKENO-DAR (Direct Angular Representation) code has been developed for the criticality safety analysis.

Before the execution of MKENO-DAR code, it should be emphasized that the MKENO-DAR code can not be executed without pre-stage computations and complex data handlings.

The codes and the utility programs are provided for the pre-stage computations and data handlings, and these codes, utility programs and relevant libraries span the MKENO-DAR CODE SYSTEM. The MKENO-DAR CODE SYSTEM is widely spanned a whole code system of MKENO-DAR code and the code system contains 3 computation codes, 5 data handling utility programs, and 5 data libraries in 3 data types.

This manual is provided to show the operating method and structure of the system without discuss the micro-scopic computations such as the calculation method, expressions, and output results. The manual will covers the only important features and input datas of the codes and utility programs. The descriptions for the output messages are omitted, since they are trivial.

The second chapter discusses the overview of the MKENO-DAR CODE SYSTEM.

The third chapter discusses the utility programs.

The fourth chapter discusses the first pre-stage computation code MAIL-DAR.

The fifth chapter discusses the second pre-stage computation code REMAIL-DAR.

The sixth chapter discusses the MKENO-DAR code.

The seventh chapter discusses the job control cards of the codes and utility programs.

The appendices show the useful informations such as the references.

2. MKENO-DAR CODE SYSTEM

2.1 The STRUCTURE OF THE MKENO-DAR CODE SYSTEM

The MKENO-DAR CODE SYSTEM is consisted of 3 codes, 5 utility

1. INTRODUCTION

Improving the Monte-Carlo code MULTI-KENO, the MKENO-DAR (Direct Angular Representation) code has been developed for the criticality safety analysis.

Before the execution of MKENO-DAR code, it should be emphasized that the MKENO-DAR code can not be executed without pre-stage computations and complex data handlings.

The codes and the utility programs are provided for the pre-stage computations and data handlings, and these codes, utility programs and relevant libraries span the MKENO-DAR CODE SYSTEM. The MKENO-DAR CODE SYSTEM is widely spanned a whole code system of MKENO-DAR code and the code system contains 3 computation codes, 5 data handling utility programs, and 5 data libraries in 3 data types.

This manual is provided to show the operating method and structure of the system without discuss the micro-scopic computations such as the calculation method, expressions, and output results. The manual will covers the only important features and input datas of the codes and utility programs. The descriptions for the output messages are omitted, since they are trivial.

The second chapter discusses the overview of the MKENO-DAR CODE SYSTEM.

The third chapter discusses the utility programs.

The fourth chapter discusses the first pre-stage computation code MAIL-DAR.

The fifth chapter discusses the second pre-stage computation code REMAIL-DAR.

The sixth chapter discusses the MKENO-DAR code.

The seventh chapter discusses the job control cards of the codes and utility programs.

The appendices show the useful informations such as the references.

2. MKENO-DAR CODE SYSTEM

2.1 The STRUCTURE OF THE MKENO-DAR CODE SYSTEM

The MKENO-DAR CODE SYSTEM is consisted of 3 codes, 5 utility

programs and 5 data libraries of 3 different types of libraries.

The codes, utility programs and the libraries are shown as table 2.2.1.

Table 2.1.1 THE COMPOSITION OF THE MKENO-DAR CODE SYSTEM

CODES	UTILITY PROGRAMS	LIBRARIES
MAIL-DAR	INIT	DOUBLE DIFFERENTIAL CROSS SECTION LIB.
REMAIL-DAR	MTCOPY	
MKENO-DAR	COPY	MGCL-LIB. (26-GR)
	DATACV	(137-GR)
	TREE	SMF-LIB. (26-GR)
		(137-GR)

NOTE: Both MGCL and SMF libraries have two energy group libraries in each, one is 26 energy group and the other is 137 energy group. These MGCL and SMF libraries are already registered in NEA DATA BANK.

2.2 SCHEMATIC CHART OF THE MKENO-DAR CODE SYSTEM

The schematic chart is provided to show the overview of the MKENO-DAR CODE SYSTEM. Fig. 2.2.1 shows that the execution of the MKENO-DAR code is the final stage of the calculation process. The great elaboration is required until the creation of macroscopic cross section data as a input of the MKENO-DAR code. This elaboration can not be avoided by the reason of certain restrictions in the input, output data handlings.

2.3 RESTRICTIONS IN DATA HANDLINGS AND DATA POOL

The restrictions are caused on the type of the data strage. Three types of the data strage are used in the MKENO-DAR CODE SYSTEM such as the EBCDIC TYPE, BINARY TYPE, and specially provided direct access data strage named "DATA POOL".

The "DATA POOL" has certain structural form inside and this structural form refuse the applications of usual manner of data handlings and substitution by any other type of data strage. By this reason, data pool handling utility programs are provided.

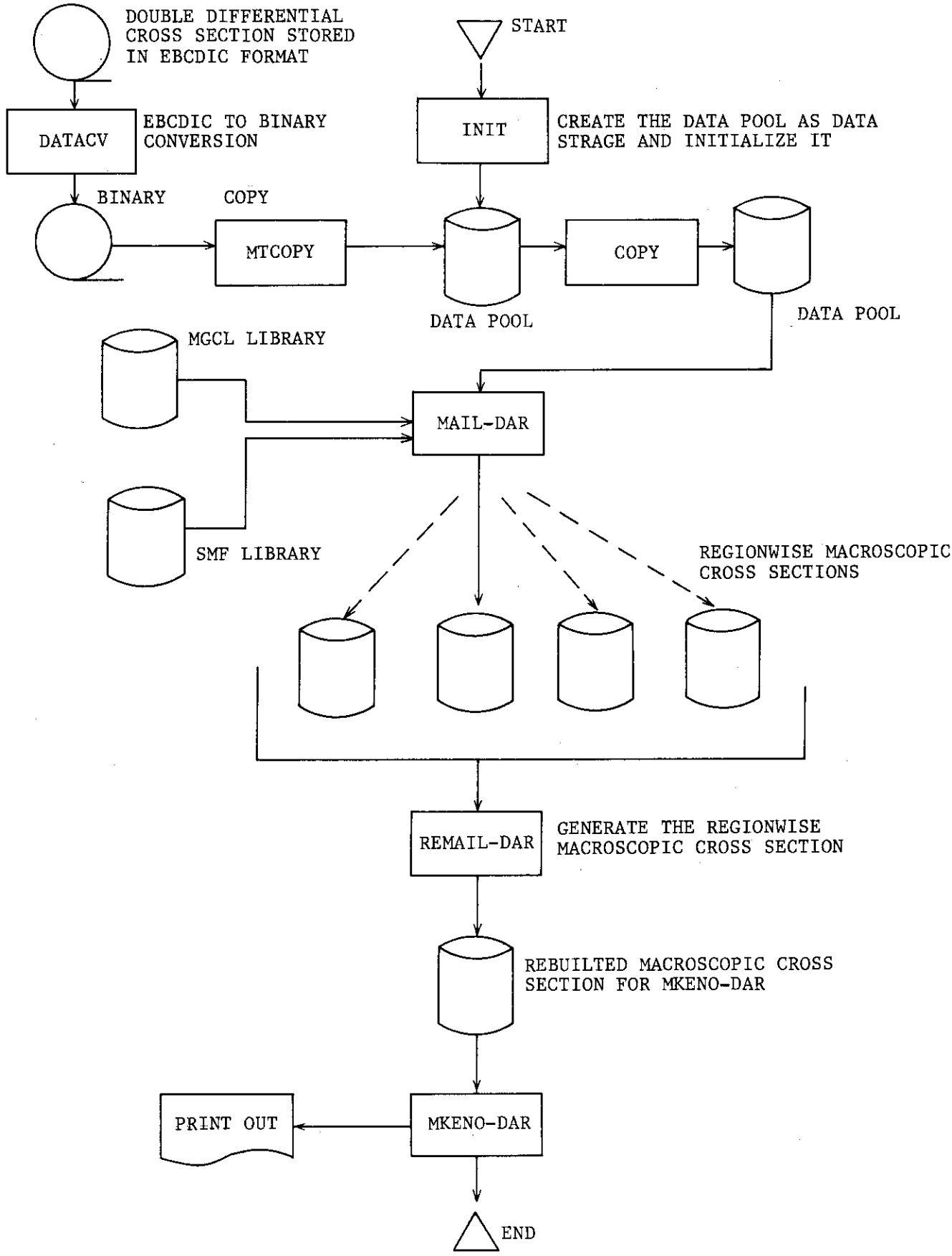

Fig. 2.2.1 Schematic chart of the MKENO-DAR CODE SYSTEM

2.4 UTILITY PROGRAMS

All the utility programs showing table 2.1.1 except "DATACV" program are the data handling programs of the data pool. Only the "DATACV" is the format conversion program of the double differential cross section data between "EBCDIC" to "BINARY" format or vice versa. The "DATACV" program can not accept the data pool as the input or the output.

2.5 THE COMPUTING PROCESS

The computing process through out the start to end in the MKENO-DAR CODE SYSTEM can be devided into several steps by the means of data handlings and computations. By the consideration to avoide the complexity of the system, the discussion of the whole system is expressed in terms of the discussion of each step.

The user of MKENO-DAR CODE SYSTEM should follow the step by step from step 1, since, the discussion pursue the actual calculation process of the MKENO-DAR CODE SYSTEM.

2.5.1 STEP 1 (THE CREATION OF THE DATA POOL)

The special direct access data strage "DATA POOL" can be generated and initialized by using the utility program "INIT". This generated "DATA POOL" is used as the data strage for the double differential cross section data transfered by using the utility program "MTCOPY" expressed in step 3.

The process of data pool generation is as follows.

① create the file on disk with next attributions.
LOGICAL RECORD LENGTH (LRECL) = 3600,
BLOCK SIZE = 3600,
RECORD FORMAT = F,
DISORG = PS
SPACE REQUIREMENT = (100,20) TRACKS (Approx.)

② Allocate the file unit "FT91F001" to the file created by ①.

③ Execute the "INIT" program.
"INIT" program reforms the file created in ① to a data pool.

2.5.2 STEP 2 (THE DATA CONVERSION FROM EBCDIC TO BINARY)

The double differential cross section datas of 41 nuclides are provided and stored in EBCDIC format in the data library "MKENO-DAR-LIB". The nuclides in the MKENO-DAR-LIB are shown in table 2.5.1.

The EBCDIC formatted datas should be converted into "BINARY" format through the utility program "DATACV" for the nuclides which the computation desired.

The datas in EBCDIC format are provided only for the visual check of the datas, and they are meaningless at the computational side of view.

At the computational side of view, the computer code "MAIL-DAR", which is the first code of the computation, requires the double differential cross section in a data pool. However, the datas in EBCDIC format can not be transfered into a data pool directly. By this reason, the BINARY type of data should be created as the middle point of the EBCDIC to a data pool data transference. (fig. 2.5.1)

The attribution of BINARY data file is shown in follows,

LOGICAL RECORD LENGTH = 6208
BLOCK SIZE = 6212,
RECORD FORMAT = VBS
DISORG = PS
SPACE REQUIREMENT = (100,20) TRACKS (Approx.)

NKENO-DAR-LIB

Table 2.5.1 The nuclides in double differential cross section library

	NUCLIDES	RCD. SIZE	BLK. SIZE	MODE
1	PU238. DATA	80	32000	EBCD1C
2	PU239. DATA	80	32000	EBCD1C
3	PU240. DATA	80	32000	EBCD1C
4	PU241. DATA	80	32000	EBCD1C
5	PU242. DATA	80	32000	EBCD1C
6	U233. DATA	80	32000	EBCD1C
7	U234. DATA	80	32000	EBCD1C
8	U235. DATA	80	32000	EBCD1C
9	U236. DATA	80	32000	EBCD1C
10	U238. DATA	80	32000	EBCD1C
11	AL27. DATA	80	32000	EBCD1C
12	B10. DATA	80	32000	EBCD1C
13	B11. DATA	80	32000	EBCD1C
14	CR. DATA	80	32000	EBCD1C
15	C12. DATA	80	32000	EBCD1C
16	C2. DATA	8	32000	EBCD1C
17	F. DATA	80	32000	EBCD1C
18	FE. DATA	80	32000	EBCD1C
19	GD. DATA	80	32000	EBCD1C
20	H1. DATA	80	32000	EBCD1C
21	SI. DATA	80	32000	EBCD1C
22	ZIRC2. DATA	80	32000	EBCD1C
23	MN55. DATA	80	32000	EBCD1C
24	MO. DATA	80	32000	EBCD1C
25	NI. DATA	80	32000	EBCD1C
26	N14. DATA	80	32000	EBCD1C
27	O16. DATA	80	32000	EBCD1C
28	PB. DATA	80	32000	EBCD1C
29	MG. DATA	80	32000	EBCD1C
30	CO59. DATA	80	32000	EBCD1C
31	HE3. DATA	80	32000	EBCD1C
32	HE4. DATA	80	32000	EBCD1C
33	BE9. DATA	80	32000	EBCD1C
34	NA23. DATA	80	32000	EBCD1C
35	CL. DATA	80	32000	EBCD1C
36	K. DATA	80	32000	EBCD1C
37	CA. DATA	80	32000	EBCD1C
38	CU. DATA	80	32000	EBCD1C
39	KR80. DATA	80	32000	EBCD1C
40	ZR90. DATA	80	32000	EBCD1C
41	TC99. DATA	80	32000	EBCD1C

2.5.3 STEP 3 (THE DATA TRANSFER FROM BINARY FILE TO DATA POOL)

This is the final step of the data transference from EBCDIC to a data pool as noted in step 2.

The created BINARY data of double differential cross section data is transfered into a data pool by using the utility program "MTCOPY".

Allocate the unit FT01F001 to the input data file such as the BINARY data, and unit FT91F001 to the data pool created by step 1.

The flow of the double differential cross section data is shown in fig. 2.5.1.

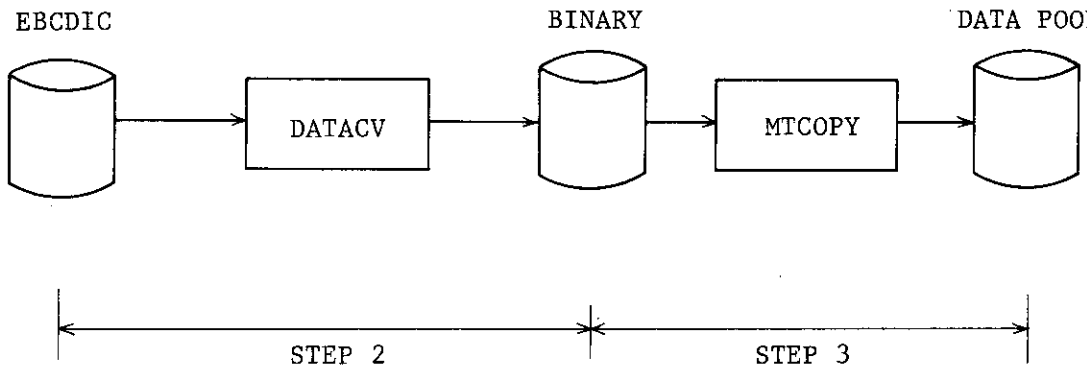

Fig. 2.5.1 Data transference from EBCDIC to a DATA POOL

2.5.4 STEP 4 (THE CREATION OF MACROSCOPIC CROSS SECTION)

It is emphasized that the macroscopic cross section is created from the microscopic cross sections of certain number of basic nuclides. For instance, the macroscopic cross section of H_2O is created from the microscopic cross sections of H and O.

By repeating step 1 to step 3, a certain number of double differential cross sections are generated and stored in the number of data pools. However, the first computation code "MAIL-DAR" requires all microscopic cross sections relative to one macroscopic cross section should be stored in one data pool, since "MAIL-DAR" can not accepts the multi data pool allocation at once.

By the reason of above, the double differential cross section files in terms of data pools should be recollected in one data pool by using the utility program "COPY".

The "copy" program works as shown in fig. 2.5.2.

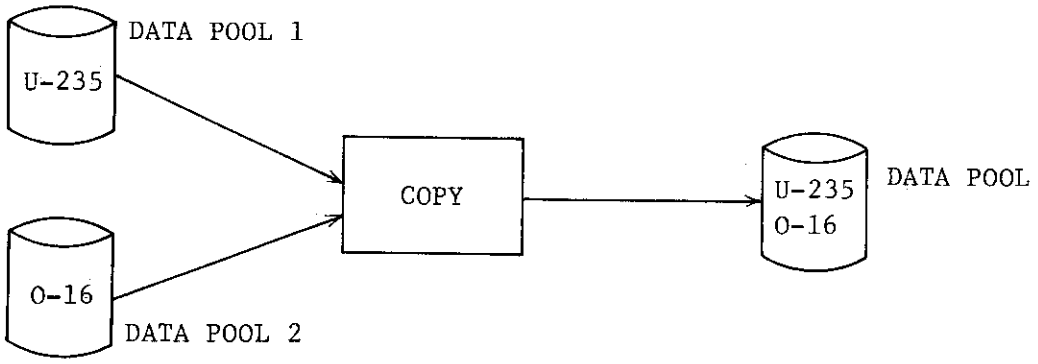

Fig. 2.5.2 Systematic chart of copy program

2.5.5 STEP 5 (THE EXECUTION OF MAIL-DAR CODE)

The "MAIL-DAR" code generates a regionwise macroscopic cross section data. Execute the "MAIL-DAR" code with a data pool which is created by step 4 and both "MGCL", "SMF" libraries as inputs.

It is obvious that the data pool created in step 4 is a source data of one region of number of regions. In other words, N region requires N source datas, if all regions are constituted with different nuclides. Create the regionwise macroscopic cross sections by repeating step 1 to step 5 until all regions are satisfied.

The output data of "MAIL-DAR" code is normal "BINARY" data, hence, the output file is choosed usual VBS file not a data pool.

Systematic chart is given as fig. 2.5.3.

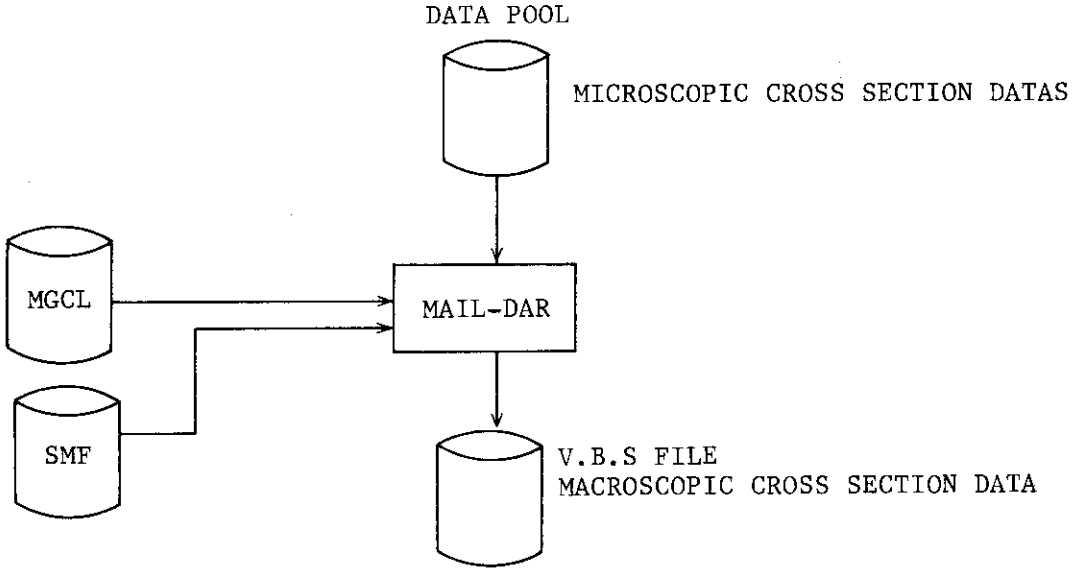

Fig. 2.5.3 Systematic chart of MAIL-DAR

2.5.6 STEP 6 (THE EXECUTION OF REMAIL-DAR CODE)

By repeating step 1 to step 5, number of macroscopic cross section datas are generated. However, the "MKENO-DAR" code can not accept the multi data allocation of macroscopic cross section datas at once. By this reason, the reconstruction of macroscopic cross section datas should be made by "REMAIL-DAR" code.

Execute the "REMAIL-DAR" code, then the number of macroscopic cross section datas are collapsed and reorganized into one macroscopic cross section data.

"REMAIL-DAR" code can be allocated maximum 40 regionwise macroscopic cross section datas in once, and the allocation of macroscopic cross sections should be made by the order of unit increase from FT10F001 to FT49F001. The output results are written in a normal VBS file, not a data pool.

The systematic chart is given as fig. 2.5.4.

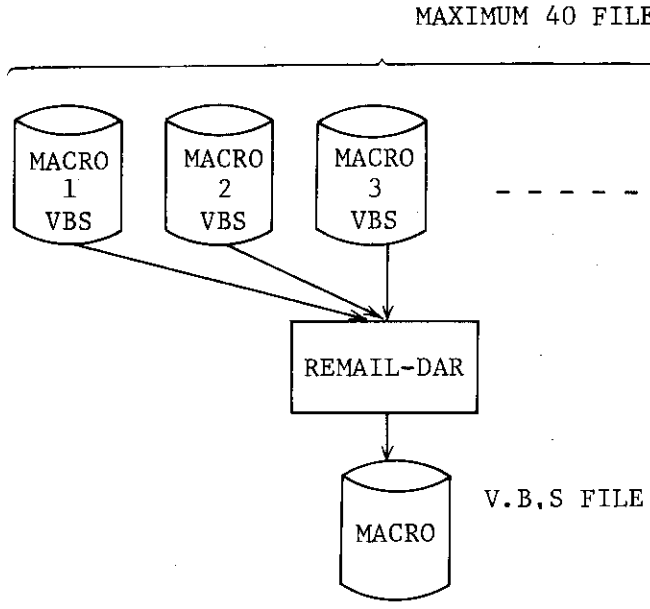

Fig. 2.5.4 Systematic chart of REMAIL-DAR

2.5.7 STEP 7 (THE EXECUTION OF MKENO-DAR CODE)

This is the final step of the computation process. The computation result of criticality safety analysis is obtained by execution of "MKENO-DAR" code with rebuilted macroscopic cross section data generated by step 6 as input.

3. UTILITY PROGRAMS OF MKENO-DAR CODE SYSTEM

The feature and the input datas of the utility programs are discussed in this chapter.

3.1 INIT PROGRAM

The data pool initialization is executed by using "INIT" program. The initialization means erasing the all datas from the cataloged data set and certain logical structure is placed in the data set. The cataloged data set should be created by "JOB CONTROL CARDS" or the commands of "VISUAL DISPLAY" with logical recod length is 3600 bytes, block size is 3600, and record format is F. "INIT" program initialize the data set and rebuilt it to a data pool.

A sample job control cards are shown in chapter 7.3.

The input data of "INIT" program is as follows.

CARD1 (FREE FORMAT)

N : THE DIRECTORY SIZE

CARD2 (FORMAT = 16A4)

TITLE : THE TITLE OF THE DATA POOL

3.2 MTCOPY PROGRAM

The "MTCOPY" program transfers and converts the binary data from magnetic tape or sequential data set to a data pool.

This program is only applicable for the binary data of double differential cross section, and the data pool initialization should be prepared by "INIT" program before the execution of MTCOPY program.

No input data required when the execution.

A sample job control cards are provided in chapter 7.4.

3.3 TREE PROGRAM

"TREE" program displays the tree structure of the data pool. It is easily found that what the datas inside the data pool.

No input datas are required.

A sample job control cards are shown in chapter 7.5.

3.4 COPY PROGRAM

The copy of the data from a data pool to another data pool is performed by using the "COPY" program. When "*ALL" is entered as a input, all datas contained in a data pool are copied to another. If the name of a certain data which shown by execution of "TREE" program is entered, the corresponding data is copied. If the same name is already exist in the proposed data pool, the "COPY" program is terminated without copying.

The name of the data in the data pool is defined as follows. By the execution of "TREE" program, the tree structure of the data pool is displayed like fig. 3.1.

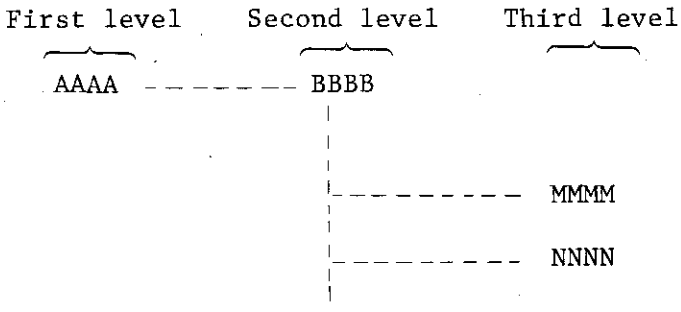

Fig. 3.1 Example of the structure of a data pool. Each node names are expressed maximum 4 characters.

The correct name of data MMMM and NNNN are AAAA·BBBB·MMMM and AAAA·BBBB·NNNN alternatively. The name AAAA, BBBB, MMMM, ···, are called node name, and AAAA is called the first node name, and BBBB is called second node name, and MMMM, NNNN are called third node names, and so on. The definition of the correct node name is due to the shortest path through out the first level to Nth level which the proposed node is existed. If AAAA·BBBB·MMMM is choosed as a input of "copy" program, the data MMMM is only copied to another data pool. If AAAA or AAAA·BBBB is entered the data MMMM and NNNN are copied. If "*ALL" is entered, the result is exactly same as AAAA is entered.

Input data for the "copy" program is shown as follows.

CARD1 : (FORMAT = A80)

CORRESPONDING NODE NAME OR *ALL

The sample job control cards are shown in chapter 7.6.

3.5 DATACV PROGRAM

The data conversion from EBCDIC format to BINARY format or BINARY format to EBCDIC format is performed by "DATACV" program. This program is executable only if the data is double differential cross section.

The input datas of "DATACV" program are shown as follows.

CARD1 (FORMAT = A1,4X,I1)

CHCK : OPERATION FLAG

= E EBCDIC to BINARY

= B BINARY to EBCDIC

IPRING : PRINT OPTION OF NODE INFORMATION

= 0 NO PRINT

= 1 PRINT

The sample job control cards are shown in chapter 7.7.

4. MAIL-DAR CODE

The "MAIL-DAR" code is developed for the MKENO-DAR CODE SYSTEM improving the "MAIL" code by adding a function, such as the calculation of macroscopic cross section using the cumulative distribution function.

The major functions of "MAIL-DAR" code are as follows.

① The calculation of the macroscopic cross section for the MKENO-DAR code.

② Analytic handling of the hydrogen scattering.

③ The calculation of the atomic number density.

4.1 SCHEMATIC CHART OF MAIL-DAR CODE

The schematic chart of "MAIL-DAR" code is given in fig. 4.1.

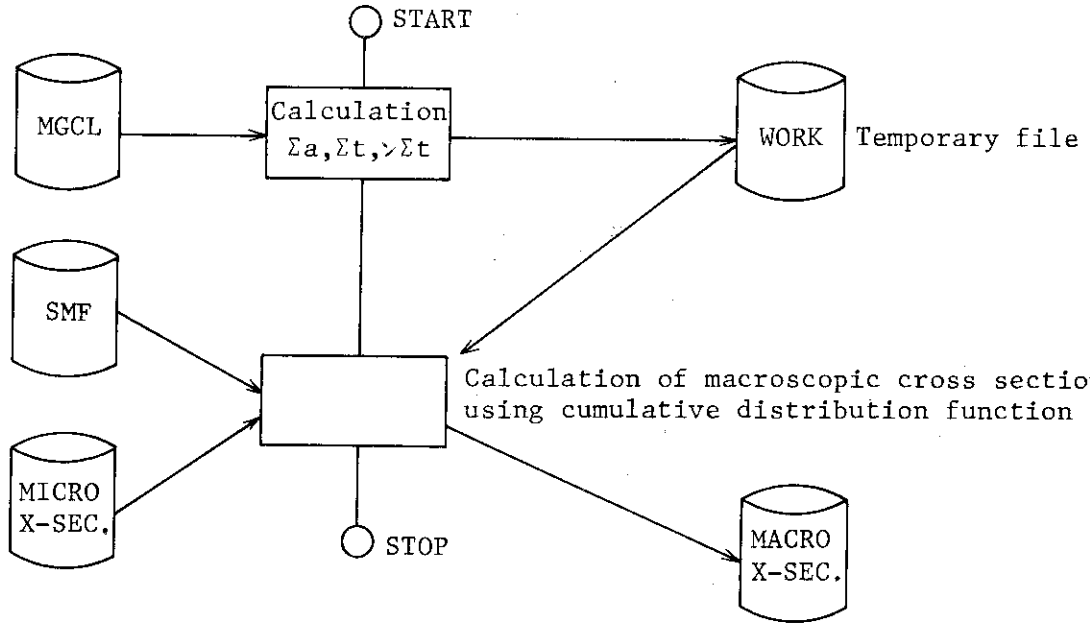

Fig. 4.1 Schematic chart of MAIL-DAR

4.2 INPUT INSTRUCTIONS OF MAIL-DAR

CARD 1 : FORMAT (10I5) CONTROL CARD

COLUMN			
1 - 5	MAXREG	:	NUMBER OF REGIONS IN THE SYSTEM
6 - 10	IGM	:	NUMBER OF ENERGY GROUPS (137 OR 26)

11－15 IPRINT : PRINT OPTION OF Σ_{eff} LIBRARY
= 0 NO PRINT
= 1 PRINT

16－20 IFSPEC : FISSION SPECTRUM OPTION
= 1 FISSION SPECTRUM OF U^{235}
= 2 FISSION SPECTRUM OF Pu^{239}
≠ 1 OR 2 FISSION SPECTRUM OF COMMON WATER MODERATOR IS EMPLOYED.

- MAXREG should be choosed always 1, since, the input data which the data pool contains the data of a region.

CARD 2 : FORMAT (2I5) OPTION FOR COLLAPSING THE SCATTERING ANGLES

COLUMN

1－5 NE : NUMBER OF COLLAPSED SCATTERING ANGLES

6－10 MM : NUMBER OF ANGULAR MESH (64, 32, 24, OR 16)

- MM should be choosed the equal number to the angular mesh of the double differential cross section.

CARD 3 : FORMAT (10I5) OPTION FOR THE EACH DIVISION PER ENERGY GROUP

(NG(i), NAN(i), i=1,NE)

where NG(i) = UPPER BOND OF ENERGY GROUP NUMBER.
NAN(i) = NUMBER OF SCATTERING ANGLE DIVISION.

- NG(i-I)+1 group to NG(i) group are the cumulative distribution function of NAN(i) divisor.
- NAN(i) are the common divisor of MM.
- NG(i) is increased simply and NG(NE)=IGM.

CARD 4 : FORMAT (20A4) TITLE CARD

(ITITLE(i), i=1,20) : TITLE OF THE CALCULATION

- The first 4 character of the title, namely ITITLE(1), is used as a parameter expressed as follows,

ITITLE(1) = ATM1 : Calculate the atomic number density of UO_2, PuO_2, or PuO_2-UO_2 automatically.

ITITLE(1) = ATM2 : Calculate the atomic number density of $UO_2(NO_3)_2$ automatically.

ITITLE(1) = ATM3 : Calculate the atomic number density of $Pu(NO_3)_4$ automatically.

If ITITLE(1) is differ to above, atomic number density should be given by cards.

CARD 5 : FORMAT (4I5, F10.0)

COLUMN		
1 - 5	NMAX	: NUMBER OF NUCLIDES IN THE REGION. If ITITLE(1) ≠ ATM1, ATM2, or ATM3, choose NMAX=0.
6 - 10	IREG	: DANCOFF CORRECTION OPTION = 0 NO EFFECTS = 1 SLAB ARRAY = 2 RECTANGULAR LATTICE OF SYLINDERS =-2 TRIANGULAR LATTICE OF SYLINDERS
11 - 15	IPRINT	: PRINT OPTION = 0 PRINT MICROSCOPIC CROSS SECTIONS, F-TABLE AND G-TABLA
16 - 20	NSTOP	: NUMBER OF ITERATIVE SIGMA-0 CALCULATIONS CUT OFF (*1). DEFAULT VALUE = 10
21 - 30	EPSL	: SIGMA-0 CONVERGENCE CRITERION VALUE (*2). DEFAULT VALUE = 0.0001

CARD 6 : FORMAT 6(5X,I7)

(JNAME(i), i=1,NMAX) : NUCLIDE ID. NUMBER IN THE REGION

NUCLIDE ID. number is constructed from 7 integers and each column indicates the following meanings.

In the case of $^{235}_{92}U$ ID. = 4 9 2 2 3 5 0 (7 columns)

① ② ③ ④

① The mark of source data which is ENDF/B-4

② Number of PROTONS

③ Mass number (If natural element, triple naught (000) should be given.)

④ Meaningless column, 0 should be entered.

This card is required, only if ITITLE(1) ≠ ATM1, ATM2, or ATM3.

CARD 7 : FORMAT (6E12.0)

(DEN(i), i=1,NMAX) ATOMIC NUMBER DENSITY CORRESPOND TO JNAME.
[a/barn·cm]

This card is required, only if ITITLE(1) ≠ ATM1, ATM2, or ATM3.

CARD 8 : FORMAT (4I5)

COLUMN			
1 - 5	IAW	:	UNIT OPTION OF RATIO OF COMPONENT NUCLIDE
			≦ 0 UNIT = [a/o]
			> 0 UNIT = [w/o]
6 - 10	ITD	:	SET ALWAYS = 1
11 - 15	NUO	:	INDICATOR OF NATURAL URANIUM OR NOT
			≦ 0 NATURAL URANIUM
			> 0 ENRICHED URANIUM
16 - 20	MIS	:	SET ALWAYS = 0

This card is required, only if ITITLE(1) = ATM1.

CARD 9 : FORMAT (3F10.0)

COLUMN			
1 - 10	PUI	:	PLUTONIUM ENRICHMENT [a/o] OR [w/o] UNIT
11 - 20	DTD	:	DENSITY OF FUEL [g/cm^3]
			+, - SIGN REQUIRED
21 - 20	OM	:	OXIGEN RATIO OF $2UO_2$ [OXIGEN/METAL]

This card is required, only if ITITLE(1) = ATM1.

CARD 10 : FORMAT (3F10.0,2I5)

COLUMN			
1 - 10	DEN	:	THE DENSITY OF THE $UO_2(NO_3)_2$ SOLUTION [g/cm^3]
11 - 20	CONCEN	:	THE CONCENTRATION OF THE URANIUM IN THE SOLUTION [gU/cm^3]
21 - 30	ACID	:	EXCESS NITRIC ACID [mol/ℓ]
31 - 35	IAW	:	SAME AS CARD 8
36 - 40	NUO	:	SAME AS CARD 8

This card is required, only if ITITLE(1) = ATM2.

CARD 11 : FORMAT = (3F10.0,I5)

COLUMN			
1 - 10	DEN	:	THE DENSITY OF THE $Pu(NO_3)_4$ SOLUTION [g/cm^3]
11 - 20	CONCEN	:	THE CONCENTRATION OF THE PLUTONIUM IN THE SOLUTION [gPu/cm^3]
21 - 30	ACID	:	EXCESS NITRIC ACID [mol/ℓ]
31 - 35	IAW	:	SAME AS CARD 8

This card is required, only if ITITLE(1) = ATM3.

CARD 12 : FORMAT (4F10.0)

COLUMN			
1 - 10	U34	:	THE RATIO OF U^{234} IN THE URANIUM [a/o] OR [w/o]
11 - 20	U35	:	THE RATIO OF U^{235} IN THE URANIUM [a/o] OR [w/o]
21 - 30	U36	:	THE RATIO OF U^{236} IN THE URANIUM [a/o] OR [w/o]
31 - 40	U38	:	THE RATIO OF U^{238} IN THE URANIUM [a/o] OR [w/o]

This card is required, only if ITITLE(1) = ATM1 or ATM2.

CARD 13 : FORMAT (7F10.0)

COLUMN			
1 - 10	P38	:	THE RATIO OF Pu^{238} IN THE PLUTONIUM [a/o] OR [w/o]
11 - 20	P39	:	THE RATIO OF Pu^{239} IN THE PLUTONIUM [a/o] OR [w/o]
21 - 30	P40	:	THE RATIO OF Pu^{240} IN THE PLUTONIUM [a/o] OR [w/o]
31 - 40	P41	:	THE RATIO OF Pu^{241} IN THE PLUTONIUM [a/o] OR [w/o]
41 - 50	P42	:	THE RATIO OF Pu^{242} IN THE PLUTONIUM [a/o] OR [w/o]
51 - 60	A41	:	THE RATIO OF Am^{241} IN THE PLUTONIUM [a/o] OR [w/o]
61 - 70	PN37	:	THE RATIO OF Np^{237} IN THE PLUTONIUM [a/o] OR [w/o]

This card is required, only if ITITLE(1) = ATM1 or ATM3.

CARD 14 : FORMAT (3F12.0)

COLUMN			
1 - 12	THFUEL	:	FUEL THICKNESS (cm)
13 - 24	THMODL	:	MODERATOR THICKNESS (cm)
25 - 36	CROSS1	:	MACROSCOPIC TOTAL CROSS SECTION OF MODERATOR AT EPI-THERMAL ENERGY REGION (cm^{-2}) (*3) FOR THE CASE OF H_2O, CROSS1=1.487

This card is required, only if IREG = 1 of card 5.

CARD 15 : FORMAT (4F12.0)

COLUMN			
1 - 12	PITCH	:	FUEL LATTICE PITCH (cm)
13 - 24	PELETR	:	FUEL PELLET RADIUS (cm)
25 - 36	FUELR	:	FUEL CLAD OUTER RADIUS (cm)
37 - 48	CROSS1	:	SAME AS CARD 14

This card is required, only if IREG = 2 of CARD 5.

- Card 4 to 15 should be repeated MAXREG times.

NOTE

(*1) : Default value = 10

(*2) : Default value = 0.0001

(*3) : For water moderator 1.487 (cm^{-1}) is recommended.

5. REMAIL-DAR CODE

The "REMAIL-DAR" code collapsies the regionwise macroscopic cross section datas created by "MAIL-DAR" code, and regenerates the macroscopic cross section for "MKENO-DAR" code. Systematic chart is given as fig. 5.1.

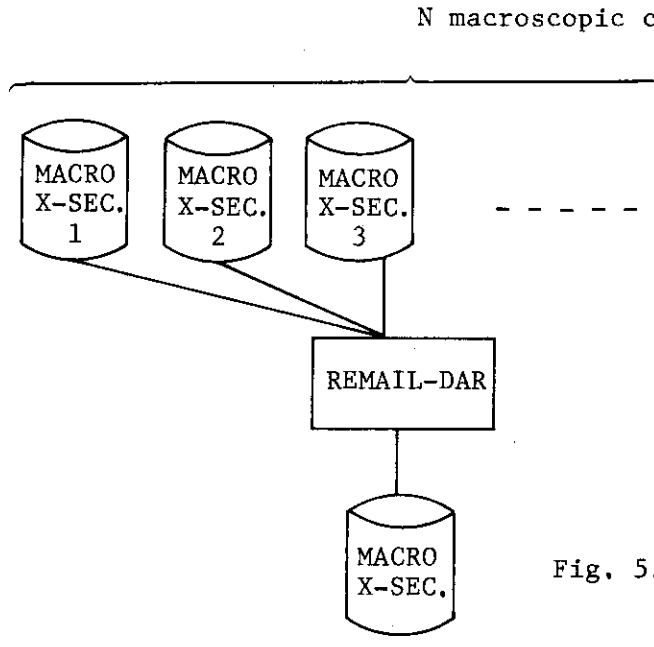

Fig. 5.1 Systematic chart of REMAIL-DAR

5.1 INPUT INSTRUCTIONS OF REMAIL-DAR

CARD 1 FORMAT = (4I5)

COLUMN			
1 - 5	IGM	:	NUMBER OF ENERGY GROUPS
6 - 10	IRG	:	TOTAL NUMBER OF THE MACROSCOPIC CROSS SECTIONS IN FILES
11 - 15	IFL	:	NUMBER OF THE MACROSCOPIC CROSS SECTION FILES
16 - 20	IPRINT	:	PRINT OPTION (1≦IPRINT≦40)

CARD 2 FORMAT = (14I5)

(NFL(i), i=1,IFL) : NUMBER OF THE MACROSCOPIC CROSS SECTION IN EACH FILE.

CARD 3 FORMAT = (14I5) REPEATED IFL TIMES

(IDRG(j,i), j=1,NFL(i)) : NEW REGION IDENTIFICATION NUMBER IF ZERO ARE ENTERED, THE CARD IS SKIPPED.

6. MKENO-DAR CODE

6.1 Free form input data program

MKENO-DAR allows data to be entered in an unformatted manner by separating each data item by one or more blanks. All 80 columns of any card may be used, and data, with certain exceptions noted below, can start or end in any column. Decimal data may be entered as in FORTRAN input, e.g., 1.733-4, 1.733E-4 or 0.0001733, is the same as 1.733×10^{-4}. Note that no imbedded blanks are allowed within a given number representation. Since blank are ignored, all zeros are must be entered.

Geometry description words, such as SUPER BOX, CYLINDER, SPHERE, etc., must begin in column 1 of a card and be separated by two or more blanks from the rest of the free-form data on the card. Additional card(s) following the geometry description word may be used, with the data in any columns 1-80 inclusive. Each new geometry description word must start in column 1 of a new card.

MKENO-DAR has provisions for multiple entries of the same data value. This is done by entering the number of repeats, following by either R, *, or $, followed by the data value to be repeated. For example, 5R2 or 5*2 enters five successive 2's in the input data. There should be no blanks between the number of repeats and the repeat flag (R, *, or $), but each multiple entry must be separated from the rest of the data by 1 or more blanks. Multiple zeros may be specified as NZ where N is the number of zeros to be repeated. There should not be any blanks between the N and Z but the NZ must be separated from the rest of the data by one or more blanks.

Certain data items such as cross-section decks, fission spectra, and albedoes are entered in free form. Proper formats for these items are given in the data guide. The title card contains identification information only and no data. The END CASE and END KENO cards must be start in column 1 and can not contain any data.

An END CASE card is really a flag to signal the end of data for a given problem. This is particularly useful if one problem in set of stacked cases contains an error, because it helps prevent the code from reading into the next problem. Once the END CASE card is encountered, the program knows it has finished with the problem, whether or not it encountered all the expected data, and it immediately prepares to read

the data for a new problem. It should be noted that, if one problem expects to utilize data from the preceding problem, they must not be separated by an END CASE card. Most errors encountered during the tracking procedure are presumed to be programming errors and result in termination of execution rather than continuing on to the next problem.

The END KENO card causes the program to cease excution.

6.2 INPUT INSTRUCTIONS OF MKENO-DAR

Card 1 Title card. FORMAT(20A4) Contains title only.

Card(s) 2 Parameter card. (Parameters are separated by one or more blanks. A new card may be started after any parameter,)

1	TMAX	Maximum computer time (in minutes) to be allowed for problem, or for each iteration if a search is to be made.
2	NBA	Number of generations.
3	NPB	Number of neutrons per generation.
4	NSKIP	Number of generations to be skipped. If fixed source problem option is used, NSKIP is ignored.
5	NGP	Number of energy group.
6	NDS	Number of downscatters or energy transfers (includes inscatter).
7	NMAT	Number of input cross-section set.
8	MATT	Number of mixtures.
9	NMIX	Number of mixing table entries. (see Card(s) 6)
10	KREFM	Total number of geometry cards. This includes the regions generated by the automatic reflector option, the CELL BOUNDARY card, and the CORE BOUNDARY card, whether calculated by the automatic reflector option, or entered separately. Do not count the REFLECTOR card from the automatic reflector option and do not count SUPER BOX card and BOX TYPE cards. All other geometry cards must be included.
11	SBOX	The number of SUPER BOX types. NOTE : SBOX must be zero for a single unit. A single unit is a configuration that does not have to be enclosed in a cube or cuboid and can not be stacked into array.

12 SBXMAX Number of units in the x direction of the super box array. A value must be entered for a single unit problem, but it is not used.

13 SBYMAX Number of units in the y direction of the super box array. A value must be entered for a single unit problem, but it is not used.

14 SBZMAX Number of units in the z direction of the super box array. A value must be entered for a single unit problem, but it is not used.

15 NTAPE |NTAPE| is the number of input cross-section set to be read from a library. If NTAPE>0, read a MKENO-DAR cross-section library on logical unit 41. If NTAPE<0, read an AMPX working format cross-section library on logical unit 4.

16 NXX Specified albedo-k_∞ OPTIONS.
NOTE : Albedo can not be used for a single unit problem.

=0 No albedo or k_∞ to be used.

=1 Uses specular reflection (k_∞). Note that this consist of mirror image reflection, multiplying the weight (WT) by the absolute value of the reflector constant (card 4) for that face, and leaving the energy unchanged. NXX=1 can not be used for a problem that utilize both specular reflection and differential albedos.

=2 Read differential albedos from cards or page. If a combination of differential albedos and specular reflection are to be used, NXX must be 2.

=3 Use differential albedos from the previous case. Can not be used in the first case following an "END CASE" card.
NOTE : Differential albedos can not be used for an adjoint problems.

17 NSCH Search type.

=0 If no search.

=1 Search on dimensions.

=2 Search on the number of units (array search). Use only if SBOX=1 (parameter 11 or card 2).

=3 Search on dimensions using a small number of generations, NBA1 (given as parameter 4 of card 3). Once convergence

has been achieved, an additional search is made using the number of generations read in as NBA (parameter 2 of card 3) iterations. This option enables the user to minimize the hazards of a poor starting guess and yet still obtain a significant number of histories in a relatively shorter time interval than required if run using NSCH=1.

18 LIST Supplies print flags to MKENO-DAR (rour-digit number).

THOUSAND DIGIT

=0 PRINT ALL macroscopic cross sections.

=1 PRINT ONLY macroscopic 1-D cross sections.

=2 DO NOT PRINT any macroscopic cross sections.

HUNDREDS DIGIT

=0 DO NOT PRINT array unit interaction matrix. (Fission probability matrix by unit).

=1 PRINT array unit interaction matrix. Use only if MATRIX (parameter 26 of card 2) is 1 or 3.

TENS DIGIT

=0 EDIT neutron balance table for each super box and summary.

=1 EDIT neutron balance table for each outer CELL BOUNDARY plug option 0.

=2 EDIT neutron balance table for each box type plug option 1.

=3 EDIT neutron balance table for all regions and summary.

UNITS DIGIT

=0 DO NOT print input cross sections from tape.

=1 PRINT input cross sections.

NOTE : For example, assume you wish to print input cross sections, macroscopic cross sections, and the array unit interaction matrix but not region-dependent fissions and absorptions, then LIST=0101.

19 NOXS Specifies whether to reuse macro cross sections and/or the geometry description from the preceding case (two-digit number).

NOTE : NOXS must always be zero for the first case following

and END CASE card.

TENS DIGIT

=0 read new geometry.

=1 use geometry from the preceding case. However, the mixed box orientation data must be read in again if NBOX (parameter 11, card 2) is greater than 1.

UNITS DIGIT

=0 read new cross sections.

=1 use cross sections from the preceding case. If using cross sections from the preceding case, the units digit of NADJ (parameter 23, card 2) must be the same for both cases.
NOTE : For example, to use cross sections from the preceding case and new geometry, NOXS=01. To read new cross sections and to use the geometry from the preceding case, NOXS=10.

20 NTYPST The type of starting distribution to be used. NTYPST must be negative to read restart data. If NTYPST is negative, the absolute value of it specifies which set of restart data is to be used. The restart data is written sequentially on tape as described in NRSTRT, (parameter 25, card 2). Note that a problem that reads restart data consists only of a title card and parameter cares. All other data is read in from the restart units. Restart data is written on unit 45 and is read in from unit 44. Note that, whenever X, Y, and Z are used in the start information, they are actually integer position indicators that define the position of the specified unit in the array.
$1 \leqq X \leqq SBXMAX$, $1 \leqq Y \leqq SBYMAX$ and $1 \leqq Z \leqq SBZMAX$.

=0 flat over the overall dimensions, in fissile material only.

=1 cosine over the overall array dimensions, in fissile material only.
Not applicable for single-unit problems.

=2 arbitrary fraction started in fissile material in unit (X,Y,Z), the rest started in fissile material with cosine distribution, over the array, about unit (X,Y,Z). Not applicable for single-unit problems.

=3 all are started at position (x,y,z) in cm, in unit (X,Y,Z).

=4 all are started at position (x,y,z) in cm, with all units of super box type NBOXST (card 13) being equally probable. Not applicable for single-unit problems.

=5 flat distribution in fissile material in units of super box type NBOXST (card 13). Not applicable for single-unit problems.

=6 starting distribution is arbitrarily input. This is the only way neutrons can be started in the reflector of an array.

=7 applicable for fixed source problem, using distributed source. If this option is used, card 15 is needed.

=8 applicable for fixed source problem, using point source. if this option is used, card 15 is needed.

NOTE : NTYPST must be 0, 3, or 6 for a single-unit problem. If any other value is specified, it will be run as a start type zero.

21 NFIX Neutron trace flag.

<0 neutron are traced to NFIXth fission neutrons.

=0 only initial generated neutrons are treated.

>0 neutrons are traced to disappearance.

22 NFDEN Fission density flag.

=0 fission densities will be calculated.

≠0 fission densities will not be calculated.

23 NADJ Calculation flag.

=0 a forward calculation will be done.

=1 an adjoint calculation will be done.

24 NXCUTE =0, the logical device number, XSECS, is set to 41 for the MKENO-DAR cross-section library and 4 for AMPX cross-section library.

≠ 0, the logical device number for the input cross-section library, XSECS, is set equal to the absolute value of NXCUTE.

25 NRSTRT Specifies the number of generations between writing of restart data. The sets of restart data for each problem are numbered sequentially starting with 1. If NRSTRT=0, no restart data will be generated.

26 MATRIX Flag for input velocities and matrix calculations.
TENS =0, read energy and lethargy from tape and calculate
DIGIT velocities.
=1, read velocities from cards.
=2, use velocities from the previous case. Note that NADJ (parameter 23, card 2) must be the same for both cases. Note that velocities are used to calculate lifetimes and generation times. Lifetime and generation time are incorrect if a differential albedo reflector is used.

UNITS
DIGIT =0, no matrix k_{eff} will be calculated.
=1, matrix k_{eff} by array unit will be calculated.
=2, matrix k_{eff} by super box type will be calculated.
=3, matrix k_{eff} by both array unit and super box type will be calculated.
NOTE : The k_{eff} and co-factor k_{eff} will be printed. If the unit interaction matrix is to be printed, LIST (parameter 18, card 2) must include a 1 in the hundreds digit.

27 NPST Position of σ_t if ANISN format cross section are read from cards.
NPST=0 if cross sections other than ANISN format are to be used.

28 NPSGG Position of σ_{gg} if ANISN format cross sections are read from cards.
NPSGG=0 if cross sections other than ANISN format are to be used.

29 IP1 Scattering angle option.
=0 average scattering angle (P1 approximation) is used.
=1 cumulative distributed scattering angle is used.

Card 3 Search parameters. Enter only if NSCH>0. (parameter 17, card 2).

1 CONSTK The desired k_{eff} for a search problem.

2 NSIG The maximum number of standard deviations k_{eff} may be from CONSTK for search completion.

3 NUMBR If NSCH=1 or 2 (parameter 17, card 2), NUMBR is the maximum number of iterations the search will run. If NSCH=3, NUMBR is the number of iterations allowed for coarse convergence.

4 NBXMA Enter only if NSCH=2 (parameter 17, card 2). The maximum number of units that will be allowed in the X direction during an array search.

4(a) NBA1 Enter if NSCH-3 (parameter 17, card 2). The number of batches to be run to achieve coarse convergence. (See explanation for NSCH=3.)

5 NBYMA Enter only if NSCH=2 (parameter 17, card 2). Maximum number of units that will be allowed in the Y direction during an array search.

5(a) NUMBRF Enter only if NSCH=3 (parameter 17, card 2). The number of iterations to be run to achieve fine convergence. (See explanation for NSCH=3.) If coarse convergence was not achieved, fine convergence will not be attempted.

6 NBZMA Enter only if NSCH=2 (parameter 17, card 2). Maximum number of units that will be allowed in the Z direction during an array search.

Card 4 Reflector Constants. Enter only if NXX≠0 (parameter 16, card 2).

NOTE : Reflector constants should be the POSITIVE albedo ID for the faces using DIFFERENTIAL ALBEDOS, the NEGATIVE albedo ID for faces using SPECULAR ALBEDOS, and zero for faces having no albedo treatment. SPECULAR ALBEDOS may be used on some faces and DIFFERENTIAL ALBEDOS on others in problems where the use of DIFFERENTIAL ALBEDOS has been indicated in NXX (parameter 26, card 2). The absolute value of the reflector constant entered for specular albedo is the fractional return for that face. The value of the reflector constant is the albedo ID for differential albedo.

1 REFCST(1) Reflector constant for +x face of the array.
2 REFCST(2) Reflector constant for -x face of the array.
3 REFCST(3) Reflector constant for +y face of the array.
4 REFCST(4) Reflector constant for -y face of the array.

5 REFCST(5) Reflector constant for +z face of the array.

6 REFCST(6) Reflector constant for -z face of the array.

Card(s) 5 Velocities. Enter only if the TENS DIGIT of MATRIX is equal to 1. NGP (parameter 5, card 2) entries will be read. A velocity must be entered for each energy group. The units on the velocity is cm/sec.

Card(s) 6 Mixing Table. Enter only if the units digit of MOXS=0 (parameter 19, card 2).

1 KKA Mixture number. It must lie between 1 and MATT (parameter 8, card 2).

2 NMA Nuclide ID number. A negative nuclide ID number indicates that the fission spectrum for that nuclide will be used for mixture KKA. A negative nuclide ID MUST be specified in each mixture that contains fissionable material.

3 RHOA Number density (atom/barn-cm); must be greater than zero.

Repeat, starting with KKA, for each nuclide. Each set of KKA, NMA, RHOA is a mixing table entry.

NOTE : There must be NMIX (parameter 9, card 2) sets of entries.

Card(s) 7 Cross Sections from Cards. Enter only if units digit of NOXS=0 (parameter 19, card 2) and |NTAPE|<NMAT (parameter 15 and 7, card 2). There will be NMAT-|NTAPE| cross-section decks entered. Each cross-section deck consists of the card sequence 7(a), 7(b), and 7(c) described below.

NOTE : Cross sections must be formatted.

Card(s) 7-a Title Card. Format (17A4, A3, I1).
Cols. 1-71 XST Nuclide identification.
Cols. 72 IORDER Enter 0 if P0 component only; enter 1 if P1 is present.

Card(s) 7-b P0 and P1 cross-section sets.
First enter the P0 component for all energy groups.
Next, if IORDER=1 (Card 7-2), enter the P1 component for all energy groups.

NOTE : Both the P0 and P1 components must be entered in either MKENO-DAR, KENO IV or ANISN format described under A and B below. All cross-section sets from cards must be entered

in the same format for a given problem.

A. MKENO-DAR or KENO FORMAT, see next page.

B. ANISN FORMAT, enter only if NPST≠0 (parameter 27, card 2).

Card(s) 7-c Fission Spectrum. FORMAT (6E12.5) Enter only if $\nu\sigma_f \neq 0$ for at least one energy group. There must be NGP entries. WARNING: Cross sections and fission spectrum cannot be read in free-form format.

Card(s) 8 Geometry Cards and Weights. Enter only if the tens digit of NOXS=0 (parameter 19, card 2). Starting in column 1 on a new card, enter the geometry word, followed by at least two blanks. Then the mixture number, dimensions and weights are entered, separated by one or more blanks. This information nay be carried over to a new card after any entry. Note that the geometry type must ALWAYS start in column 1. A weight for each energy group must follow each geometry card (except SUPER BOX, BOX TYPE or REFLECTOR cards, which are not counted as geometry cards). If SBOX≧1 (parameter 11, card 2), SUPER BOX cards are needed, and start in column 1 and punch SUPER BOX, followed by two or more blanks. Then enter the super box type. If NBOX>1 (See card(s) 8-a) BOX TYPE cards are needed, and start in column 1 and punch BOX or BOX TYPE, followed by two or more blanks ; if NBOX≦1, BOX TYPE cards may not be entered. This card is followed by as many geometry cards and weights as are necessary to describe the box type. Repeat this process until all box types have been described. If SBOX≧1 (parameter 11, card 2), enter a CELL BOUNDARY card next. The CELL BOUNDARY cards start in column 1 and punch CELL BDY, followed by two or more blanks. The mixture field contains a zero and the remainder of the field is punched with cuboid dimensions that fit tightly around the array of box types. Following the CELL BDY card are weights for each energy group. The remaining super box regions are described as any appropriate geometry type, in the manner illustrated under Card(s) 8-c. Repeat this process until all super box types have been described.

NOTE : ALL REGIONS WITHIN A GIVEN BOX TYPE MUST BE DESCRIBED SO THAT EACH SUCCESSIVE REGION COMPLETELY ENCLOSES THE

PREVIOUS REGION. THE ADJACENT FACES OF BOXES, WHICH ARE CONTAINED BY SAME SUPER BOX, IN CONTACT WITH EACH OTHER MUST BE THE SAME SIZE.

If an external reflector to an array is present, enter a CORE BOUNDARY card. The CORE BOUNDARY card starts in column 1 and the first four characters must be CORE followed by two or more blanks. The word CORE may be followed by a blank and the word BDY or BOUND, which then must be followed by two or more blanks. The mixture field contains a zero and the remainder of the field is punched with cuboid dimensions that fit tightly around the array. Following the CORE card are weights for each energy group. The remaining reflector regions are described as any appropriate geometry type, in the manner illustrated under Card(s) 8-b.
NOTE : EACH SUCCESSIVE REFLECTOR REGION MUST COMPLETELY ENCLOSE THE PREVIOUS REGION. A weight for each energy group must follow each reflector region card.

The REFLECTOR card starts in column 1 and says REFLECTOR; followed by two or more blanks followed by a mixture number. The next six entries indicate the desired reflector thickness on each face (+x, -x, +y, -y, +z, and -z, respectively). The reflector thickness must be either zero or positive. They cannot be negative. Following the thickness is the ID number of the weights to be read from tape. If the ID is less then 10, the weights will be read from cards as given in card(s) type 10. The REFLECTOR card may be replace the CORE BOUNDARY card or be placed at any point external to it. If it replaces the CORE BOUNDARY card, it calculates the core boundary, supplies the weights for it, and fills in the reflector regions and their associated weights until KREFM-1 (parameter 10, card 2) regions contain data. It then fills the last region with the remaining reflector thickness and supplies the weights associated with it. The thickness of each region is governed by data associated with the weights read from tape or cards (3 cm for water and paraffin, 5 cm for concrete, etc.). If KREFM (parameter 10, card 2) is too large so the maximum reflector thickness is used up before

reaching KREFM regions, it simply pads with zero thickness regions until it accumulates KREFM regions. If the REFLECTOR card is external to the CORE BOUNDARY card, it follows the same procedure except it does not calculate the core boundary but starts creating regions at the point where the REFLECTOR card was read.

NOTE : The first automatic reflector region always uses the weights for the first increment (i.e., 0-3 cm for water and paraffin, 0-5 cm for concrete, etc.). Therefore, exercise caution in choosing weights for any regions that occur between the CORE BOUNDARY card and the REFLECTOR card. If the REFLECTOR card replaces the CORE BOUNDARY card, you need not be concerned.

Card(s) 8-a Super Box Type Card. If SBOX=0, (parameter 11, card 2), do not enter a SUPER BOX TYPE Card.

Start in Col. 1 "SUPER BOX" (left adjusted).

1 I Super box number (between 1 and SBOX), starting two or more spaces after the geometry card.

2 NBOX Number of box types contained by SUPER BOX 1.

3 NBXMAX Number of units in the x direction of the array made by box types.

4 BNYMAX Number of units in the y direction of the array made by box types.

5 NBZMAX Number of units in the z direction of the array made by box types.

Card(s) 8-b Box Type Card. If NBOX=0, (parameter 11, card 2), do not enter a Box Type Card. If NBOX=1, a Box Type Card may be entered but is not necessary.

Start in Col. 1 "BOX TYPE" (left adjusted).
Starting two or more spaces after the geometry word, enter the box number (between 1 and NBOX).

Card(s) 8-c Geometry Cards.
NOTE : All geometry words start in Col. 1.

FGEOM FGEOM may be one of the following and must be left adjusted: CUBE, CUBOID, SPHERE, CYLINDER, XCYLINDER, YCYLINDER, HEMISPHERE, HEMISPHE+Z, HEMISPHE-Z, HEMISPHE+X, HEMISPHE-X, GENERAL, XHEMICYL+Y, XHEMICYL-Y, HEMISPHE+Y, HEMISPHE-Y, XHEMICYL+Z, XHEMICYL-Z, YHEMICYL+X, YHEMICYL-X, YHEMICYL+Z, YHEMICYL-Z, ZHEMICYL+X, ZHEMICYL-X, ZHEMICYL+Y, ZHEMICYL-Y, CORE BDY, CELL BDY, REFLECTOR.
NOTE : FGEOM may be no more than 12 characters long.

CUBE has +X = +Y = +Z and -X = -Y = -Z. Note that the +X dimension need not equal the -X dimension of the cube; i.e., the origin need not be at the center of the cube.

CUBOID is a rectangular parallelepiped and may be described anywhere relative to the origin.

SPHERE must be centered about the origin.

CYLINDER has its length described along the Z axis and its center line must lie on the Z axis.

XCYLINDER has its length described along the X axis and its center line must lie on the X axis.

YCYLINDER has its length described along the Y axis and its center line must lie on the Y axis.

HEMISPHERE must have its flat portion centered about the origin at Z=0.0 and exists only in the positive Z direction.

HEMISPHERE(B)(C) must have its flat portion centered about the origin at (c)=0.0 and exists only in the BC direction. (B = + or -, C = X, Y, or Z). For example, HEMISPHE+Z is the same as the previously described HEMISPHERE and HEMISPHE-Z is the mirror image of HEMISPHE+Z, therefore existing only in the negative Z direction.

(B)HEMICYL(C)(D) is a half cylinder whose axis is the B axis (B = X, Y, or Z) and exists only in the CD direction (C = + or -, D = X, Y, or Z). (Examples: ZHEMICYL+X, YHEMICYL-Z, XHEMICYL+Y).

GENERAL refers to generalized geometry. A GENERAL card must be entered for each generalized geometry media. The

purpose of the GENERAL card is to set up a correspondence between each medium number and a mixture. The first GENERAL card should contain the MKENO-DAR mixture number corresponding to medium 1 as defined in GEOM, the second should contain the MKENO-DAR mixture number for medium 2, etc. The dimension specification portion of the GENERAL cards may be set to zero. MKENO-DAR automatically inserts the outer ZONE boundaries from the generalized geometry data for the dimensions of the last GENERAL card. Note that for a single-unit problem (SBOX=0, parameter 11, card 2) a GENERAL card can be the last card entered. However, for an array problem the last card must be a cube or cuboid whose dimensions are as large or larger than the zero dimensions of the generalized geometry region.

Starting two or more spaces after the geometry word, the following data is entered, separated by one or more blanks. A new card may be started after any entry.

MAT	Mixture number. (enter a zero for a void)
XX(1)	Radius for sphere, cylinders, hemispheres, hemicylinders, +x dimension for cube, cuboid, or general region.
XX(2)	-x dimension for cube, cuboid, or general region, +z for cylinder, +x for x cylinder, +y for y cylinder, + length for hemicylinder, omit XX(2) for a sphere or hemisphere.
XX(3)	+y dimension for cuboid or general region, -z for dylinder, -x for x cylinder, -y for y cylinder, - length for hemicylinder, omit XX(3) for a sphere, hemisphere, or cube.
XX(4)	-y dimension for cuboid or general region, omit for all other geometry types except CORE BDY.
XX(5)	+z dimension for cuboid or general region, omit for all other geometry types except CORE BDY.
XX(6)	-z dimension for cuboid or general region, omit for all other geometry types except CORE BDY and CELL BDY.

Weights

WTAVG The weight which is given a neutron that survives Russian roulette.
Enter a value for each energy group. Enter a weight of 0.5 or 0.0 for all regions within the core. If a value of 0.0 is entered it is defaulted to 0.5 within the code.
Weights for some commonly used reflector materials are given in ORNL-TM-4660.

Repeat the card(s) 8-a, 8-b, 8-c sequence until SBOX super box types, each super box contains NBOX box types, have been described. Boxes must be numbered sequentially starting with 1.
NOTE : The last geometry card for each box type must be a cube or cuboid.

Card(s) 8-c0 Cell Boundary Card (must be cuboid). Enter 'CELL BDY' for putting an external cell around BOX TYPE array in each SUPER BOX. Each SUPER BOX is enclosed by CELL BOUNDARY similar to the CORE BOUNDARY of KENO-IV. A CELL BOUNDARY defines outline of the array consisted by BOX TYPEs. Usually a SUPER BOX is used together with CELL BOUNDARY card. If only single SUPER BOX is used, CELL BOUNDARY card is not needed.

Card(s) 8-c1 Core Boundary Card (must be cuboid). Enter only if there are additional regions external to the core. This card is needed only if one more of cards 8-c2 are used.

Starting in Col. 1.

CORE BDY (left adjusted).

MAT Enter a mixture number (usually zero) ; leave two or more blanks between CORE BDY and MAT.

XX(1) +x dimension for a cuboid or cube.

XX(2) -x dimension for a cuboid or cube.

XX(3) +y dimension for a cuboid, zero for a cube.

XX(4) -y dimension for a cuboid, zero for a cube.

XX(5) +z dimension for a cuboid, zero for a cube.

XX(6) -z dimension for a cuboid, zero for a cube.

NOTE : These dimensions must fit tightly around the array.

WTAVG Enter a value for each energy group even through are not used.

Card(s) 8-c2 Reflector geometry Cards. See card(s) 8-c.

FGEOM

MAT

XX(1).....XX(6)

Weights

Repeat the above card sequence until all reflector regions have been described.

Card(s) 8-c3 Automatic Reflector Card (must be cuboid). This card can be entered in the place of a core boundary card (in which case it calculates the core boundary and fills in any remaining regions) or it may be entered at any point external to the core boundary card (it then just fills in any remaining cuboidal regions). It must never be internal to any other geometry type. Use only one Automatic Reflector card per problem.

NOTE : All regions generated by the automatic reflector option must be counted in KREFM (card 2, parameter 10).

Starting in Col. 1.

REFLECTOR (left adjusted) Enter only if the automatic reflector option is to be exercised for putting an external reflector around an array.

MAT Enter the mixture number of the material comprising the reflector ; leave two or more blanks between REFLECTOR and MAT.

XX(1) The reflector thickness in the +x direction. Non-negative numbers only.

XX(2) The reflector thickness in the -x direction. Non-negative numbers only.

XX(3) The reflector thickness in the +y direction. Non-negative numbers only.

XX(4) The reflector thickness in the -y direction. Non-negative numbers only.

XX(5) The reflector thickness in the +z direction. Non-negative numbers only.

XX(6) The reflector thickness in the -z direction. Non-negative numbers only.

IDWT The ID of the appropriate set of WTs to be read from library. If IDWT is less than 10, the weights will be read from cards. (See card(s) 10.)
NOTE : Do not enter weights for the REFLECTOR card. They are automatically provided through reading IDWT.

Card(s) 9-a Mixed Box Orientation Card. Enter only if NBOX>1. (parameter 2, card 8-a).
The first field contains the box type, followed by three sets of three fields that are treated like FORTRAN DO loops, followed by a field that indicates whether another set of mixed box data is to be read. The arrangement of boxes may be considered as consisting of a three-dimensional matrix of box type numbers, with the box position increasing in the positive X, Y, and Z directions, respectively. Each set of mixed box orientation data consists of the following parameters, separated by one or more blanks.

LTYPE The box type. LTYPE must be greater than zero and less than or equal to NBOX (parameter 2, card 8-a).

IX1 The starting point in the x direction. IX1 must be at least 1 and less than or equal to NBXMAX (parameter 3, card 8-a).

IX2 The ending point in the x direction. IX2 must be at least 1 and less than or equal to NBXMAX.

INCX The number of boxes by which increments are made in the positive x direction. INCX must be greater than zero and less than or equal to NBXMAX.

IY1 The starting point in the y direction. IY1 must be at least 1 and less than or equal to NBYMAX (parameter 4, card 8-a).

IY2 The ending point in the y direction. IY2 must be at least 1 and less than or equal to NBYMAX.

INCY The number of boxes by which increments are made in the positive y direction. INCY must be greater than zero and less than or equal to NBYMAX.

IZ1 The starting point in the z direction. IZ1 must be at

least 1 and less than or equal to NBZMAX (parameter 5, card 8-a).

IZ2 The ending point in the z direction. IZ2 must be at least 1 and less than or equal to NBZMAX.

INCZ The number of boxes by which increments are made in the positive z direction. INCZ must be greater than zero and less than or equal to NBZMAX.

ISTP Indicates whether to read another set of mixed box orientation data.
=0, read another set of data.
≠0, do not read any more mixed box orientation data.

An important feature of this type of data description is that if any portion of an array is defined in a conflicting manner, the last card to define that portion will be the one that determines the array's box type configuration. To utilize this feature, one can bill an entire array with the most prevalent box type and then superimpose the other box types in their proper places to accurately describe the array. The last set of mixed box orientation data must have a nonzero entry in the last field.
Repeat this card until box orientation of all super box have been described. This card must be entered in super box type sequence.

Card(s) 9-b Mixed Super Box Orientation Cards. Enter only if SBOX>1 (parameter 11, card 2).
This card must be entered in the same manner as card 9-a.

LTYPE The super box type. LTYPE must be greater than zero and less than or equal to SBOX (parameter 11, card 2).

IX1 The starting point in the x direction. IX1 must be at least 1 and less than or equal to SBXMAX (parameter 12, card 2).

IX2 The ending point in the x direction. IX2 must be at least 1 and less than or equal to SBXMAX.

INCX The number of super boxes by which increments are made in the positive x direction. INCX must be greater than zero and less than or equal to SBXMAX.

IY1 The starting point in the y direction. IY1 must be at

least 1 and less than or equal to SBYMAX (parameter 13, card 2).

IY2 The ending point in the y direction. IY2 must be at least 1 and less than or equal to SBYMAX.

INCY The number of super boxes by which increments are made in the positive y direction. INCY must be greater than zero and less than or equal to SBYMAX.

IZ1 The starting point in the z direction. IZ1 must be at least 1 and less than or equal to SBZMAX (parameter 14, card 2).

IZ2 The ending point in the z direction. IZ2 must be at least 1 and less than or equal to SBZMAX.

INCZ The number of super boxes by which increments are made in the positive z direction. INCZ must be greater than zero and less than or equal to SBZMAX.

ISTP Indicates whether to read another set of mixed super box orientation data.
=0, read another set of data.
≠0, do not read any more mixed super box orientation data.

The data of this card have same features as card 9-a. The last set of mixed super box orientation data must have a nonzero entry in the last field.

Card(s) 10 Reflector Weights from Cards. Enter only if IDWT (card 8-c3) is less than 10.

WTTITL Name of material being used for the reflector weights. Enter in Cols. 1-12.

IDWTT Weight ID number (usually the ID number that will be put on the library, but may be anything. The code automatically sets equal to the value of IDWT (card 8-c3)).

ISUBST Number of sets of weights associated with this IDWTT. Usually one since you need read in only 1 set of weights.

THICK The thickness in cm of each weighting region or interval.

NUMINC The number of intervals in the set of weights. (NUMINC*THICK=maximum thickness for which weights are given.)

NGPWT The number of energy groups for this set of weights. It must be equal to NGP.

WTAVG(I,J) The weight average for each interval and energy group. I=1 to NUMINC and J=1 to NGPWT. There are NUMINC*NGPWT entries.
NOTE : If SUBST is greater than 1, the data "THICK" through "WTAVG(I,J)" must be repeated ISUBST times.

Card(s) 11 Generalized Geometry Description, if any, as described in section 5.4.

Card(s) 12 Albedo Deck. Enter only if NXX=2 (parameter 16, card 2).
NOTE : ALBEDO data must be formatted. It consists of the following data :
(1) ATITLE format (18A4) Title card for the albedo deck.
(2) (WTCOS(I),I=1,NANG) format (6E12.5). WTCOS is the product of the fractional solid angle and the cosine of the polar angle for each polar angle. NANG is the number of polar angles and for the existing KENO Albedos is 4.
(3) (PLIM(I),I=1,NANG) format (6E12.5). PLIM is the cosines of the angular bounds for each of the polar angles.
(4) (CPOL(I),I=1,NANG) format (6E12.5). CPOL is the cosines of the polar angles.
(5) (SPOL(I),I=1,NANG) format (6E12.5). SPOL is the sines of the polar angles.
(6) ((((A(I,J,K,L),L=1,NANG),J=1,NGP),K=1,NANG),I=1,NGP) format (18A4). A(I,J,K,L) is the albedo data in hexidecimal form and represents the relative angular return tables for each input angle and energy.

Card 13 Data for special start options. Input if NTYPST≧2 (parameter 20, card 2).
(Entered in free form.)

if NTYPST = 2 (parameter 20, card 2) Enter the X, Y, and Z coordinates (in terms of boxes) of the box about which the starting distribution is given.

NBXS The X index of the box.

NBYS The Y index of the box.

NBZS The Z index of the box.

FX The fraction of neutrons to be started as spike in box (NBXS,NBYS,NBZS) of the array.

if NTYPST = 3 (parameter 20, card 2) Enter the X, Y, and Z indices

(in terms of boxes) of the box where the neutrons will be started as a spike at the coordinates x, y, and z in that box.

NBXS	The X index of the box.
NBYS	The Y index of the box.
NBZS	The Z index of the box.
TFX	The x coordinate of the spike in box (X, Y, and Z).
TFY	The y coordinate of the spike in box (X, Y, and Z).
TFZ	The z coordinate of the spike in box (X, Y, and Z).
if NTYPST = 4	(parameter 20, card 2) Enter the box type in which the neutrons will be started at the point (x, y, and z).
NBOXST	The box type in which the neutrons will be started.
TFX	The x coordinate of the point at which the neutrons will be started in the box type NBOXST.
TFY	The y coordinate of the point at which the neutrons will be started in box type NBOXST.
TFZ	The z coordinate of the point at which the neutrons will be started in box type NBOXST.
if NTYPST = 5	(parameter 20, card 2) Enter the box type in which the neutrons will be started.
NBOXST	The box type in which the neutrons will be started.
if NTYPST = 6	(parameter 20, card 2)
LFIN	The final neutron to be started at this point. The first LFIN must be ≧ 1. (The first neutron at this point is the one following the previous LFIN.) Points are read until LFIN = NPB (parameter 3, card 2), the number of neutrons per batch.
NBXS	The X index of the super box.
NBYS	The Y index of the super box.
NBZS	The Z index of the super box.
TFX	The x coordinate of the point.
TFY	The y coordinate of the point.
TRZ	The z coordinate of the point.

For example, assume there are 50 neutrons in a generation, and you wish to start the first five neutrons in super box (1,1,1) at x=1.0, y=0.0, z=0.0; the next 25 neutrons in box (1,2,1) at x=1.0, y=0.0, z=2.0; and the remaining neutrons

in super box (1,2,2) at x=1.5, y=1.5, z=1.5. Then the input card could be entered as follows:

5 1 1 1 1.0 0.0 0.0 30 1 2 1 1.0 0.0 2.0 50 1 2 2
1.5 1.5 1.5

Card(s) 14 Search Constants. Enter only if NSCH = 1, 2, or 3.
The physical significance of a search constant may best be described as a proportionality constant. For a dimension search, the search constant (CONS) is proportional to the relative change in dimension $(XX_{new}-XX_{old})/XX_{old}$ divided by the k-effective $(k_{new}-k_{old})$ where XX_{old} is the dimension that yielded a k-effective of k_{old} and XX_{new} is the dimension that yielded a k-effective of k_{new}. The search constants is positive if k-effective increases as the dimension increases and negative if k-effective decreases as the dimension increases.

if NSCH = 1 or 3 (parameter 17, card 2) Enter one set for each geometry region, and in corresponding order. There will be one entry on a card for a sphere or hemisphere, three entries for a cylinder, xcylinder, or ycylinder, and six entries for a cube, cuboid or general region.
Each entry corresponds to a dimension and tells how that dimension will be altered. A value of zero means that dimension will be unchanged.
NOTE : Zeros should ALWAYS be entered for a general region because a search cannot be made for a general region.

CONS(1) Search constant for the radius of a sphere, hemisphere, cylinder, xcylinder, ycylinder, hemicylinder, +x dimension of cube or cuboid.

CONS(2) Search constant for +z of cylinder, +x of xcylinder, +y of ycylinder, +x of xhemicylinder, +y of yhemicylinder, +z of zhemicylinder, -x dimension of cube or cuboid.

CONS(3) Search constant for -z of cylinder, -x of ycylinder, -y of ycylinder, -x of xhemicylinder, -y of yhemicylinder, -z, of zhemicylinder, +y dimension of cube of cuboid.

CONS(4) Search constant for -y dimension of cube or cuboid.

CONS(5) Search constant for +z dimension of cube or cuboid.

CONS(6) Search constant for -z dimension of cube or cuboid. NOTE : If NSCH=1 or 3 and the problem contains a reflector that is to maintain its thickness even if the unit spacing changes, simply enter zero for all six search constant for each of the core boundary and reflector regions. The code will automatically calculate the new core boundary and maintain proper reflector thickness and weightings.

if NHCH = 2 (parameter 17, card 2) Enter only one set. There will be three entries; one for each coordinate direction of the array. The number of units in a given direction will be changed by an integer multiple of the search constant specified. For any array search, the search constant for a given direction represents the minimum number of units by which the array size can be changed in that direction. The change in the number of units in each direction maintains the proportionality of the search constants stated for those directions. The search constant is positive if k-effective increases as the array size increases and negative if k-effective decreases as the array size increases.

CONS(1) Search constant for changing the number of units in the X direction.

CONS(2) Search constant for changing the number of units in the Y direction.

CONS(3) Search constant for changing the number of units in the Z direction.

Card 15 Parameters for fixed source problem. Enter only if NTYPST = 7, or 8 (parameter 20, card 2).

Card(s) 15-a Fixed source spectrum data.

1 KR The region number. If KR is equal to zero, enter next card.

2 FIX(KR,NGP) The fixed source spectrum at region KR for each energy group.

Card(s) 15-b Parameters for point source. Enter only if NTYPST is equal to 8 (parameter 20, card 2).

1 FX Source ratio (%) at the point (Xs, Ys, Zs).

2 Xs The x coordinate of the point.

3 Ys The y coordinate of the point.

4 Zs The z coordinate of the point.

Repeat this card (card 14-b) until sum of FX have been equal to 100.

Card 16 Data for graphic processing.

Card(s) 16-a

NUSE <0, graphic output will not be done.
≧0, graphic output will be done. If NUSE greater than zero, NUSE characters, which is used for region (material) number in the graphic output, must be entered (card 16-b). If NUSE is equal to zero, default character will be used.

Card(s) 16-b This card(s) must be entered only if NUSE greater than zero.

ATABLE(NUSE) Enter NUSE characters which is used for region (material) numbers from 0 to NUSE-1.

Card(s) 16-c This card(s) must be entered only if NUSE is greater than zero.

ITP =0, graphic processing will be ended with above data.
>0, graphic processing will be started with following data.

IRG =0, graphic output will be done on material.
>0, graphic output will be done on region.

Card(s) 16-d Plotted area definition data. Enter only if NUSE is greater than or equal to zero, and ITP is greater than zero.

XUL The x coordinate of the upper left corner for a plotted area.

YUL The y coordinate of the upper left corner for a plotted area.

ZUL The z coordinate of the upper left corner for a plotted area.

XLR The x coordinate of the lower right corner for a plotted area.

YLR The y coordinate of the lower right corner for a plotted area.

XLR The z coordinate of the lower right corner for a plotted area.

Card(s) 16-e Plotted axis definition data. Enter only if NUSE is greater than or equal to zero, and ITP is greater than zero.

UX The direction cosine of plotted vertical axis with respect to the x axis.

UY The direction cosine of plotted vertical axis with respect to the y axis.

UZ The direction cosine of plotted vertical axis with respect to the z axis.

VX The direction cosine of plotted horizontal axis with respect to the x axis.

VY The direction cosine of plotted horizontal axis with respect to the y axis.

VZ The direction cosine of plotted horizontal axis with respect to the z axis.

Card 17 END CASE. This card is optional. It enables MKENO-DAR to read to the end of a case that contains an error and to start on a new case.

Card 18 END KENO. This card is optional and comes after the last card of the last case. No more data will be read after this card has been encountered.

6.3 MKENO-DAR General Geometry Input Data

If a GENERAL card is present in the input, then generalized GEOM data must be entered. The following GEOM input description is essentially the same as that found in ORNL-3622.

The outstanding feature of GEOM is its ability to describe multiple media may be include, while the permissible boundaries may be of any shape which can be described by quadric surfaces used singly or in combination.

The initial step in the geometric description of a system for GEOM is to enclose the entire system in a cuboid whose faces are parallel with the xy, yx, and xz coordinate planes. This cuboid is then divided into several smaller cuboids, called zones, by planes parallel to the coordinate planes and extending entirely across the system.

The zones, in turn, are then divided into smaller cuboids, called blocks, by planes again parallel to the coordinate axes but extending only across individual zones. The planes used as zone and block boundaries need not necessarily be boundaries between media; however, if a boundary between two media is a plane parallel to a coordinate plane, it is advantageous to make it a block or zone boundary. The use of the zone-block scheme allows complicated parts of the system under study to be divided into smaller blocks than may be needed for simpler regions. If the whole system is relatively simple or requires a similar description throughout, the system should be composed of one zone divided into many blocks rather than many zones of one block each.

Boundaries between media which are not also block boundaries may be any quadric surface. A quadric surface is defined by the zeros of a quadratic function, and divides all space into two regions. In one region, the function defining the surface will be positive; in the other it will be negative. Each block may contain a maximum of 32 such surfaces as medium boundaries. The surfaces will divide the block into sectors. A sector is defined as a volume positive to one set of quadric surfaces but negative to another set. Each sector must contain only one medium which may be the same as the medium in another sector. Spatial volumes containing a single medium which cannot be described by a single sector definition must be divided into two or more sectors. It is not necessary to mention every surface in the block in defining a sector. It is, in fact, more efficient to include in a sector definition only those surfaces which actually form the boundary of the sector. In addition sectors containing the same medium may overlap without error.

Care must be taken in the use of cones as quadric surfaces, since the quadratic equation describes a surface of two nappes. If as is usual, the described surface is but one napper of the cone, a block boundary through the vertex must be used to cut off the surface.

Input to GEOM. (All alphabetic input must be left-adjusted.)

Card A : Format (I5)

a. An index which is not used in KENO but must be specified as a 2.

Card B : Format [A11,5(E10.5,A1)]

This card lists the zone boundaries in increasing order along the X axis, including the boundaries of the parallelpiped enclosing the entire system. Since the number of boundaries depends upon the problem,

commas in the A1 fields separating the boundaries are used to indicate that the list continues, while the absence of a comma following the last boundary indicates that the list has ended. The all field is for the programmer's convenience and will be ignored by the code.

Card(s) B' : Format [6(E10.5,A1)]

If the number of boundaries exceeds the five allowed by the format of card B, the list is continued on as many cards B' as are required.

Card C : Format [A11,5(E10.5,A1)]

Identical with card B except that the listing is of the zone boundaries in order along the Y axis.

Card C' : Format [6(E10.5,A11)]

Identical with card B' but continues the Y axis zone boundaries.

Card D : Format [A11,5(E10.5,A1)]

Identical with card B except that the listing is of the zone boundaries in order along the Z axis.

Card D' : Format [6(E10.5,A1)]

Identical with card B' but continues the Z axis zone boundaries.

Cards E through P : Constitute a complete zone description. This set of cards must be included once for each zone.

Card E : Format (A6,I5,I5,I5)
　　a　1　m　n

a. The word ZONE.

1,m,n : Each zone is located in the system by three integers: 1, m, and n. These specify the zone as being the 1th in the X direction, the mth in the Y direction, and the nth in the Z direction. The integers 1, m, and n run from 1 to the maximum number of zones in each direction.

Card F : Format [A11,5(E10.5,A1)]

This card lists the block boundaries in this zone in increasing order along the X axis, including the boundaries of the zone.

Card(s) F' : Format [A11,5(E10.5,A1)]

This is a block list continuation card similar to card B' of the zone listing.

Cards G, G'

The same as cards F and F' except that the block boundaries along the Y axis are listed.

Cards H, H'

The same as cards F and F' except that the block boundaries along the Z axis are listed.

Cards J through P : Constitute a complete block description. This set of cards must be included once for each block in the zone.

Card J : Format (A6,I5,I5,I5)
a l m n

a. The word BLOCK.

l,m,n : Each block is located in the zone by three integers: l, m, and n. These specify the block as being the lth in the direction, the mth in the Y direction, and the nth in the Z direction, within the given zone. The integers l, m, and n run from 1 to the maximum number of blocks in each direction.

Card K : Format [A12,10(I5,A1)]
a b

a. The word MEDIA.

b. A list of the media, sector by sector, in the block. As with other lists, a comma in the A1 field indicates that the list continues; its termination is indicated by the absence of the comma. A media number of 1000 signifies an internal void, while a media number of 0 signifies an external void.

Card(s) K' : Format [12(I5,A1)]

The continuation, if required, of the medium list.

Card L : Format [A12,10(I5,A1)]
a b

a. The word SURFACES.

b. A list of the quadric surfaces appearing in the block. Commas in the A1 field indicate that the list continues; a blank indicates the end of the list. The number appearing in this list derive from the order in which the surfaces are mathematically described on card R, which will be described later in the input.

Card L' : Format [8(I5,A1)]

The continuation, if needed, of the list begun on card L.

Card M : Format [A6,18I3]
a b

a. The word SECTOR.

b. The designation of each sector with reference to its position

relative to the quadric surfaces. For every sector in the block there must be a card M, which will have as many references as there are surfaces in the block. The status of the sector is listed according to the following key:

+1 : The sector is on the positive side of the surface.

-1 : The sector is on the negative side of the surface.

0 : The surface is not needed in the definition of the sector.

The order in which each reference to a quadric surface appears on each card M must correspond to the order in which the quadric surfaces are listed on card L.

If there is only one sector in a block, cards L and M should be omitted.

Card Q : Format (^{a}I5,11A6)

a. The total number of quadric surfaces in the entire system. The alphabetic data in the A6 fields is ignored by the code.

Card R : Format [4(E10.5,^{a}A5,^{b}A1,^{c})]

Each quadric surface is described by writing the quadratic function whose zeros define the surface, in a fixed field format resembling the normal manner of writing functions. Each term in the function is specified by :

a. The coefficient of the term.

b. May be XSQ, YSQ, ZSQ (used for x^2, y^2, and z^2) XZ, YX, YZ, XY, ZX, YZ, X, Y, Z, or blank.

c. A nonblank character in this field indicates the end of the function. The next function must start on a new card.

7. JOB CONTROL CARDS (JCL)

The indexed job control cards of the linkage and a sample execution are provided for each program. The job control cards for the fortran compiler and the assembler are provided independently from the programs, since, they are trivial and the usual manner of compilation and assembling are applicable.

The linkage cards give the helpfull informations such as the built up program and reference libraries in the view point of the construction of a load module of a program.

The index numbers of the job control cards specify the I/O units and libraries.

A few pages of sample output of a sample execution of each program are also provided as the reference.

The job control cards are basically written for the user of FACOM-M380 computer, thus the user of another computer should replace the job control cards carefully to much to the own computer system.

7.1 JOB CONTROL CARDS FOR THE FORTRAN COMPILER

The job control cards for the fortran compiler are shown below. All the FORTRAN-77 programs can be applied.

```
INDEX NO.
---------
          //JCLG JOB
          // EXEC JCLG
          //SYSIN DD DATA,DLM='++'
          ************************************************************
          *        USER ACCOUNT INFORMATION CARDS ARE OMITTED         *
          ************************************************************
          //FORT    EXEC PGM=JZKAFORT,REGION=1024K,COND=(4,LT),
          //   PARM='ELM(*),OPTIMIZE(2),LINECOUNT(0),NOS,NOSOURCE,NOMAP'
          //SYSPRINT  DD SYSOUT=*,
          //           DCB=(RECFM=FBA,LRECL=137,BLKSIZE=19043)
          //SYSTERM   DD SYSOUT=*,
          //           DCB=(RECFM=FBA,LRECL=137,BLKSIZE=19043)
      1   //SYSLIN    DD DSN=J????.MAILDAR.OBJ,DISP=(NEW,CATLG,DELETE),
          //           UNIT=TSSWK,SPACE=(TRK,(30,10)),DCB=BLKSIZE=3200
      2   //SYSIN     DD DSN=J????.MAILDAR.FORT77,DISP=SHR
          ++
          //
```

INDEX REFERENCES

1 : The created object module of FORTRAN-77 source program.

2 : FORTRAN-77 source program.

7.2 JOB CONTROL CARDS FOR THE ASSEMBLER

The job control cards for the assembler are shown below. All the assembler programs can be applied.

INDEX NO.

```
          //JCLG JOB
          // EXEC JCLG
          //SYSIN DD DATA,DLM='++'
          ************************************************************
          *      USER ACCOUNT INFORMATION CARDS ARE OMITTED          *
          ************************************************************
          //ASM  EXEC    PGM=JLAX00,REGION=1024K,COND=(4,LT),
          //             PARM='LIST,BATCH,NODECK,OBJ'
     1    //SYSLIB       DD   DSN=SYS1.MACLIB,DISP=SHR
          //SYSPRINT     DD   SYSOUT=*
          //SYSUT1       DD   UNIT=WK10,SPACE=(TRK,(30,10))
          //SYSUT2       DD   UNIT=WK10,SPACE=(TRK,(30,10))
          //SYSUT3       DD   UNIT=WK10,SPACE=(TRK,(30,10))
          //SYSPUNCH     DD   SYSOUT=P
     2    //SYSGO  DD  DSN=J????.GETDCB.OBJ,DISP=(NEW,CATLG,DELETE),
          //             UNIT=TSSWK,SPACE=(TRK,(30,10)),
          //             DCB=(LRECL=80,BLKSIZE=3200,RECFM=FB,DSORG=PS)
     3    //SYSIN        DD   DSN=J????.GETDCB.ASM,DISP=SHR
          ++
          //
```

INDEX REFERENCES

1 : ASSEMBLER LIBRARY.

2 : The created object module of assembler source program.

3 : Assembler source program.

7.3 JOB CONTROL CARDS OF INIT PROGRAM

The job control cards of linkage and a sample execution of the utility program "INIT" and the index references, also the logical unit table are shown in page 51. A sample output is given in page 52.

```
INDEX NO.
---------
          //JCLG JOB
          // EXEC JCLG
          //SYSIN DD DATA,DLM='++'
          ************************************************************
          *         USER ACCOUNT INFORMATION CARDS ARE OMITTED        *
          ************************************************************
          //*********************************************** START OF LINKAGE
          //LINK  EXEC PGM=JQAL,REGION=768K,COND=(4,LT),
          //         PARM='NOMAP,LIST,LET'
      1   //SYSLIB        DD   DSN=SYS1.FORTLIB,DISP=SHR
          //SYSPRINT      DD   SYSOUT=*,DCB=(BLKSIZE=4840)
          //SYSTERM       DD   SYSOUT=*
          //SYSUT1        DD   UNIT=VIO,SPACE=(TRK,(30,10))
      2   //SYSLMOD       DD   DSN=J????.INIT.LOAD,UNIT=TSSWK,
          //         DISP=(NEW,CATLG,DELETE),SPACE=(TRK,(30,10,1),RLSE),
          //         DCB=(BLKSIZE=19069,RECFM=U)
      3   //SYSLIN        DD   DSN=J????.INIT.OBJ,DISP=SHR
          //*********************************************** END OF LINKAGE
          //*********************************************** START OF EXECUTION
          //RUN   EXEC PGM=TEMPNAME,COND=(4,LT),PARM='FLIB(ERRCUT=0)'
      4   //STEPLIB       DD  DSN=J????.INIT.LOAD,DISP=SHR
          //SYSPRINT      DD  SYSOUT=*,
          //                  DCB=(RECFM=FBA,LRECL=137,BLKSIZE=19043)
          //FT05F001      DD DDNAME=SYSIN
          //FT06F001      DD  SYSOUT=*,
          //                  DCB=(RECFM=FBA,LRECL=137,BLKSIZE=19043)
      5   //FT91F001 DD DSN=J????.DP1.DATA,DISP=(NEW,CATLG),
          //              DCB=(LRECL=3600,BLKSIZE=3600,RECFM=F,DSORG=PS),
          //              UNIT=D0950,SPACE=(TRK,(100,50))
          //SYSIN DD *
      6      100
      7     THE TITLE OF THE DATA POOL SHOULD BE ENTERED MAX 64 CHARACTERS
          ++
          //
```

INDEX REFERENCES

1 : FORTRAN LIBRARY.

2 : The created load module.

3 : The object module.

4 : The load module are same as INDEX 2.

5 : The initialized data pool by INIT program.

6 : Input data (The size of directory).

7 : Input data (The title of the data pool).

Table 7.3.1 LOGICAL UNIT TABLE OF INIT PROGRAM

LOGICAL UNITS	SPECIFICATIONS
FT05F001	INPUT DATA READ.
FT06F001	OUTPUT DATA WRITE.
FT91F001	DATA POOL ALLOCATION.

A sample output of INIT program

A sample output of the utility program "INIT" is shown below.

```
ENTER DIRECTORY SIZE
ENTER TITLE (64 CHARACTERS)

*** MESSAGE FROM PINIT ***
    NO. OF INITIALIZED RECORD IS  9350

*********   C O N T R O L   S E C T I O N   *********

     COL.

    1-18  TITLE    :

      THE TITLE OF THE DATA POOL SHOULD BE ENTERED MAX 64 CHARACTERS

      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          :        102

      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                  :        100

      29  SIZE OF THE DATA SECTION                       :       9249

      30  REAL NUMBER OF THE DIRECTORY RECORDS           :          1

      31  REAL NUMBER OF THE DATA SET RECORDS            :          0
```

7.4 JOB CONTROL CARDS OF MTCOPY PROGRAM

The job control cards of linkage and a sample execution of the utility program "MTCOPY" and the index references are shown below. The logical unit table and a sample output are shown in page 54.

```
INDEX NO.
---------
          //JCLG JOB
          // EXEC JCLG
          //SYSIN DD DATA,DLM='++'
          ************************************************************
          *         USER ACCOUNT INFORMATION CARDS ARE OMITTED        *
          ************************************************************
          //*****************************************  START OF LINKAGE
          //LINK  EXEC PGM=JQAL,REGION=768K,COND=(4,LT),
          //      PARM='NOMAP,LIST,LET'
      1   //SYSLIB        DD   DSN=SYS1.FORTLIB,DISP=SHR
          //SYSPRINT      DD   SYSOUT=*,DCB=(BLKSIZE=4840)
          //SYSTERM       DD   SYSOUT=*
          //SYSUT1        DD   UNIT=VIO,SPACE=(TRK,(30,10))
      2   //SYSLMOD       DD   DSN=J????.MTCOPY.LOAD,UNIT=TSSWK,
          //          DISP=(NEW,CATLG,DELETE),SPACE=(TRK,(30,10,1),RLSE),
          //               DCB=(BLKSIZE=19069,RECFM=U)
      3   //SYSLIN        DD   DSN=J????.MTCOPY.OBJ,DISP=SHR
      4   //              DD   DSN=J????.GETDCB.OBJ,DISP=SHR
          //*****************************************  END OF LINKAGE
          //*****************************************  START OF EXECUTION
          //RUN   EXEC PGM=TEMPNAME,COND=(4,LT),PARM='FLIB(ERRCUT=0)'
      5   //STEPLIB       DD  DSN=J????.MTCOPY.LOAD,DISP=SHR
          //SYSPRINT      DD  SYSOUT=*,
          //                  DCB=(RECFM=FBA,LRECL=137,BLKSIZE=19043)
          //FT05F001      DD DDNAME=SYSIN
          //FT06F001      DD  SYSOUT=*,
          //                  DCB=(RECFM=FBA,LRECL=137,BLKSIZE=19043)
      6   //FT91F001 DD DSN=J????.DATAPOOL.DATA,DISP=SHR
      7   //FT01F001 DD DSN=J????.U236.DATA,DISP=(OLD,PASS),
          //          LABEL=(3,SL,,IN),DCB=(DSORG=PS),VOL=SER=033014,UNIT=MT62
          ++
          //
```

INDEX REFERENCES

1 : FORTRAN LIBRARY.

2 : The created load module.

3 : The object module which the source program compiled.

4 : The object module which the assembler program GETDCB assembled.

5 : The load module as same as INDEX 2.

6 : The data pool which initialized by INIT program.

7 : The data which transfer to the data pool.

Table 7.4.1 LOGICAL UNIT TABLE OF MTCOPY PROGRAM

LOGICAL UNITS	SPECIFICATIONS
FT01F001	Allocate to the transfering data stored in VBS file.
FT05F001	INPUT DATA READ.
FT06F001	OUTPUT DATA WRITE.
FT91F001	Allocate to the data pool which initialized by INIT program.

A sample output of MTCOPY program

A sample output of the utility program "MTCOPY" is shown below.

```
 MTCOPY WAS FINISHED SUCCESSFULLY

*** INFORMATION OF DATA POOL USAGE ***
    LOGICAL UNIT NO.        =     91
    DATA SET NAME           = J????.DATAPOOL.DATA
    NO. OF WRITTEN RECORDS  =   1252
    REMAINS RECORDS         =   8077
```

7.5 JOB CONTROL CARDS OF TREE PROGRAM

The job control cards of linkage and a sample execution of the utility program "TREE" and the index references, also logical unit table are shown in page 55. A sample output is given in page 56.

INDEX NO.

```
          //JCLG JOB
          // EXEC JCLG
          //SYSIN DD DATA,DLM='++'
          ************************************************************
          *         USER ACCOUNT INFORMATION CARDS ARE OMITTED        *
          ************************************************************
          //*********************************************** START OF LINKAGE
          //LINK  EXEC PGM=JQAL,REGION=768K,COND=(4,LT),
          //       PARM='NOMAP,LET,LIST'
     1    //SYSLIB         DD   DSN=SYS1.FORTLIB,DISP=SHR
          //SYSPRINT       DD   SYSOUT=*,DCB=(BLKSIZE=4840)
          //SYSTERM        DD   SYSOUT=*
          //SYSUT1         DD   UNIT=VIO,SPACE=(TRK,(30,10))
     2    //SYSLMOD        DD   DSN=J????.TREE.LOAD,UNIT=TSSWK,
          //          DISP=(NEW,CATLG,DELETE),SPACE=(TRK,(30,10,1),RLSE),
          //          DCB=(BLKSIZE=19069,RECFM=U)
     3    //SYSLIN         DD   DSN=J????.TREE.OBJ,DISP=SHR
     4    //               DD   DSN=J????.GETDCB.OBJ,DISP=SHR
          //*********************************************** END OF LINKAGE
          //*********************************************** START OF EXECUTION
          //RUN   EXEC PGM=TEMPNAME,COND=(4,LT),PARM='FLIB(ERRCUT=0)'
     5    //STEPLIB        DD  DSN=J????.TREE.LOAD,DISP=SHR
          //SYSPRINT       DD  SYSOUT=*,
          //                   DCB=(RECFM=FBA,LRECL=137,BLKSIZE=19043)
          //FT05F001       DD DDNAME=SYSIN
          //FT06F001       DD  SYSOUT=*,
          //                   DCB=(RECFM=FBA,LRECL=137,BLKSIZE=19043)
     6    //FT91F001 DD DSN=J????.DP1.DATA,DISP=SHR
          //SYSIN DD *
          ++
          //
```

INDEX REFERENCES

1 : FORTRAN LIBRARY

2 : The created load module of TREE program.

3 : The object module which the source program compiled.

4 : The object module which the assembler program GETDCB assembled.

5 : The load module as same as INDEX 2.

6 : The data pool under inspection.

Table 7.5.1 LOGICAL UNIT TABLE OF TREE PROGRAM

LOGICAL UNITS	SPECIFICATIONS
FT05F001	NOT USED.
FT06F001	OUTPUT DATA WRITE.
FT91F001	Allocate to the data pool which intend to inspect.

A sample output of TREE program

A sample output of the utility program "TREE" is shown below.

```
                    N O D E   T R E E

TITLE OF A DATA POOL              ***
    J1446.#MKENO.DAR.LIB.DATA (EGRP,64)   U235

LENGTH OF A RECORD                        ***          900

MAXIMUN NUMBER OF THE SAME LEVEL NODE     ***           74

SIZE OF THE DIRECTORY SECTION             ***           20   (USED RECORDS     5)

SIZE OF THE DATA SECTION                  ***         3443   (USED RECORDS  1252)

REMAINS OF THE DIRECTORY SECTION          ***           15

REMAINS OF THE DATA SECTION               ***         2191

LEVEL      1       2       3       4       5       6       7       8

      J137 :  NEUTRON  AND  GAMMA-RAY  ENERGY  STRUCTURE
       I
       I-----INFX :  INFINITE DILUTION CROSS SECTION
              I
              I-----1261 : 1261 U-235  FROM ENDF/B-IV 407
                     I
                     I----- SMT :  SMOOTH CROSS SECTION
                     I
                     I----- ELA :  ELASTIC SCATTERING MATRIX
                     I
                     I----- INS :  INELASTIC SCATTERING MATRIX
                     I
                     I----- N2N :  (N,2N) SCATTERING MATRIX
       I
       I-----FX64 : ANGULAR MESH
              I
              I-----1261 : 1261 U-235 MACRO (ENDF/B) 407 300K
```

7.6 JOB CONTROL CARDS OF COPY PROGRAM

The job control cards of linkage and a sample execution of "COPY" program and the index references are shown below. The logical unit table and a sample output are given in page 58.

INDEX NO.

```
          //JCLG JOB
          // EXEC JCLG
          //SYSIN DD DATA,DLM='++'
          ************************************************************
          *         USER ACCOUNT INFORMATION CARDS ARE OMITTED        *
          ************************************************************
          //*********************************************** START OF LINKAGE
          //LINK  EXEC PGM=JQAL,REGION=768K,COND=(4,LT),
          //         PARM='NOMAP,LET,LIST'
     1    //SYSLIB       DD   DSN=SYS1.FORTLIB,DISP=SHR
          //SYSPRINT     DD   SYSOUT=*,DCB=(BLKSIZE=4840)
          //SYSTERM      DD   SYSOUT=*
          //SYSUT1       DD   UNIT=VIO,SPACE=(TRK,(30,10))
     2    //SYSLMOD      DD   DSN=J????.COPY.LOAD,UNIT=TSSWK,
          //            DISP=(NEW,CATLG,DELETE),SPACE=(TRK,(30,10,1),RLSE),
          //               DCB=(BLKSIZE=19069,RECFM=U)
     3    //SYSLIN       DD   DSN=J????.COPY.OBJ,DISP=SHR
     4    //               DD   DSN=J????.GETDCB.OBJ,DISP=SHR
          //*********************************************** END OF LINKAGE
          //*********************************************** START OF EXECUTION
          //RUN   EXEC PGM=TEMPNAME,COND=(4,LT),PARM='FLIB(ERRCUT=0)'
     5    //STEPLIB      DD  DSN=J????.COPY.LOAD,DISP=SHR
          //SYSPRINT     DD  SYSOUT=*,
          //               DCB=(RECFM=FBA,LRECL=137,BLKSIZE=19043)
          //FT05F001     DD DDNAME=SYSIN
          //FT06F001     DD  SYSOUT=*,
          //               DCB=(RECFM=FBA,LRECL=137,BLKSIZE=19043)
     6    //FT01F001 DD DSN=FT1.DATA,DISP=(NEW,PASS,DELETE),
          //  DCB=(LRECL=6208,BLKSIZE=6212,RECFM=VBS,DSORG=PS),
          //  SPACE=(TRK,(200,50)),UNIT=TSSWK
     7    //FT02F001 DD DSN=FT2.DATA,DISP=(NEW,PASS,DELETE),
          //  DCB=(LRECL=6208,BLKSIZE=6212,RECFM=VBS,DSORG=PS),
          //  SPACE=(TRK,(100,10)),UNIT=TSSWK
     8    //FT91F001 DD DSN=J????.DP1.DATA,DISP=SHR
     9    //FT92F001 DD DSN=J????.DP4.DATA,DISP=SHR
          //SYSIN DD *
    10    J137.INFX.1261
          ++
          //
```

INDEX REFERENCES

1 : FORTRAN LIBRARY.

2 : The created load module of the copy program.

3 : The object module which the source program compiled.

4 : The object module which the assembler program GETDCB assembled.

5 : The load module as same as INDEX 2.

6 : The temporary file, VBS file is required.

7 : The temporary file, VBS file is required.

8 : The data pool which the proposed datas are stored.

9 : The data pool which the destination of the datas transfer.

10 : Input data.

Table 7.6.1 LOGICAL UNIT TABLE OF COPY PROGRAM

LOGICAL UNITS	SPECIFICATIONS
FT01F001	Temporary file, VBS file required.
FT02F001	Temporary file, VBS file required.
FT05F001	INPUT DATA READ.
FT06F001	OUTPUT DATA WRITE.
FT91F001	Allocate to the data pool which the proposed datas are stored.
FT92F001	Allocate to the data pool which the destination of the datas transfer.

A sample output of copy program

A sample output of the utility program "copy" is shown below.

```
ENTER NODE NAME. IF *ALL IS ENTERD, ALL DATA IS COPIED
          ***** PFIND ERROR *****
              NODE NAME INPUT ERROR
EXIST ADD NODE NAME IN A DATA POOL J????.DP4.DATA
 NODE NAME = J137.INFX.1261.
          ***** PFIND ERROR *****
              NODE NAME INPUT ERROR
EXIST ADD NODE NAME IN A DATA POOL J????.DP4.DATA
 NODE NAME = J137.INFX.1261. SMT.
          ***** PFIND ERROR *****
              NODE NAME INPUT ERROR
EXIST ADD NODE NAME IN A DATA POOL J????.DP4.DATA
 NODE NAME = J137.INFX.1261. ELA.
          ***** PFIND ERROR *****
              NODE NAME INPUT ERROR
EXIST ADD NODE NAME IN A DATA POOL J????.DP4.DATA
 NODE NAME = J137.INFX.1261. INS.
          ***** PFIND ERROR *****
              NODE NAME INPUT ERROR
EXIST ADD NODE NAME IN A DATA POOL J????.DP4.DATA
 NODE NAME = J137.INFX.1261. N2N.

*** INFORMATION OF DATA POOL USAGE ***
    LOGICAL UNIT NO.        =     92
    DATA SET NAME           = J????.DP4.DATA
    NO. OF WRITTEN RECORDS  =     49
    REMAINS RECORDS         =   2638
DATA COPY WAS FINISHED SUCCESSFULLY
ENTER NEXT NODE NAME
```

7.7 JOB CONTROL CARDS OF DATACV PROGRAM

The job control cards of the linkage and a sample execution of the utility program "DATACV" and the index references are shown below. The logical unit table and a sample output are given in page 60 to 61.

```
INDEX NO.
---------
        //JCLG JOB
        // EXEC JCLG
        //SYIN DD DATA,DLM='++'
         ************************************************************
         *         USER ACCOUNT INFORMATION CARDS ARE OMITTED        *
         ************************************************************
         //****************************************** START OF LINKAGE
         //LINK  EXEC PGM=JQAL,REGION=768K,COND=(4,LT),
         //       PARM='NOMAP,LIST,LET'
    1    //SYSLIB        DD   DSN=SYS1.FORTLIB,DISP=SHR
         //SYSPRINT      DD   SYSOUT=*,DCB=(BLKSIZE=4840)
         //SYSTERM       DD   SYSOUT=*
         //SYSUT1        DD   UNIT=VIO,SPACE=(TRK,(30,10))
    2    //SYSLMOD       DD   DSN=J????.DATACV.LOAD,UNIT=TSSWK,
         //          DISP=(NEW,CATLG,DELETE),SPACE=(TRK,(30,10,1),RLSE),
         //          DCB=(BLKSIZE=19069,RECFM=U)
    3    //SYSLIN        DD   DSN=J????.DATACV.OBJ,DISP=SHR
         //****************************************** END OF LINKAGE
         //****************************************** START OF EXECUTION
         //RUN   EXEC PGM=TEMPNAME,COND=(4,LT),PARM='FLIB(ERRCUT=0)'
    4    //STEPLIB       DD  DSN=J????.DATACV.LOAD,DISP=SHR
         //SYSPRINT      DD  SYSOUT=*,
         //                  DCB=(RECFM=FBA,LRECL=137,BLKSIZE=19043)
         //FT05F001      DD DDNAME=SYSIN
         //FT06F001      DD  SYSOUT=*,
         //                  DCB=(RECFM=FBA,LRECL=137,BLKSIZE=19043)
    5    //FT01F001 DD DSN=J????.U235.DATA,UNIT=MT62,DCB=(DSORG=PS),
         //          LABEL=(8,SL,,IN),VOL=SER=SAF090,DISP=(OLD,PASS)
    6    //FT02F001 DD DSN=J????.EBCDIC.DATA,DISP=(NEW,CATLG,DELETE),
         //          UNIT=TSSWK,SPACE=(TRK,(100,50)),
         //          DCB=(LRECL=80,BLKSIZE=3200,RECFM=FB,DSORG=PS)
         //SYSIN DD *
    7    B    1
         ++
         //
```

INDEX REFERENCES

1 : FORTRAN LIBRARY.

2 : The created load module of the DATACV program.

3 : The object module which the source program compiled.

4 : The load module as same as INDEX 2.

5 : INPUT/OUTPUT BINARY DATA.

6 : INPUT/OUTPUT EBCDIC DATA.

7 : Input data.

Table 7.7.1 LOGICAL UNIT TABLE OF DATACV PROGRAM

LOGICAL UNITS	SPECIFICATION
FT01F001	Allocate to the BINARY data set.
FT02F001	Allocate to the EBCDIC data set.
FT05F001	INPUT DATA READ.
FT06F001	OUTPUT DATA READ.

A sample output of DATACV program

The first couple of pages of print out of the utility program "DATACV" are shown below.

```
ENTER BINARY MT OR EBCDIC MT ( B/E ) ?
ENTER PRINTOPTION ( 0/1 = NON/NODE INFOMATION ) ?
FORMAT(A1,4X,I1)
CONVERSION START ( BINARY DATA ===> EBCDIC DATA )
DATE =====> 83-09-19
ICONTR ===> J1446.#MKENO.DAR.LIB.DATA (EGRP,64)   U235
ICONT ====>    900     74     10   1489      5   1319

   NTH   NADAT  NDASET
     1     12      1

   DATE    NODE(1)  NODE(2)  NODE(3)  NODE(4)  NODE(5)  NODE(6)  NODE(7)  NODE(8)  NODE(9)  NODE(10)
XX-XX-XX     J137

  INFORM(1)   INFORM(2)   INFORM(3)   INFORM(4)   INFORM(5)
        137           0           0           0           0

 ICM ===>  NEUTRON  AND  GAMMA-RAY  ENERGY  STRUCTURE

    LAST    NOARY  NDATA(1) NDATA(2) NDATA(3) NDATA(4)
     138        1       138

   NTH   NADAT  NDASET
     2     13      1

   DATE    NODE(1)  NODE(2)  NODE(3)  NODE(4)  NODE(5)  NODE(6)  NODE(7)  NODE(8)  NODE(9)  NODE(10)
XX-XX-XX     J137     INFX

  INFORM(1)   INFORM(2)   INFORM(3)   INFORM(4)   INFORM(5)
         0           0           0           0           0

 ICM ===>  INFINITE DILUTION CROSS SECTION

    LAST    NOARY  NDATA(1) NDATA(2) NDATA(3) NDATA(4)
        0        0       0

   NTH   NADAT  NDASET
     3     14      1

   DATE    NODE(1)  NODE(2)  NODE(3)  NODE(4)  NODE(5)  NODE(6)  NODE(7)  NODE(8)  NODE(9)  NODE(10)
XX-XX-XX     J137     INFX     1261

  INFORM(1)   INFORM(2)   INFORM(3)   INFORM(4)   INFORM(5)
        1261          0           0           0           0

 ICM ===> 1261 U-235  FROM ENDF/B-IV 407

    LAST    NOARY  NDATA(1) NDATA(2) NDATA(3) NDATA(4)
       10        1       10

   NTH   NADAT  NDASET
     4     15      2

   DATE    NODE(1)  NODE(2)  NODE(3)  NODE(4)  NODE(5)  NODE(6)  NODE(7)  NODE(8)  NODE(9)  NODE(10)
XX-XX-XX     J137     INFX     1261      SMT

  INFORM(1)   INFORM(2)   INFORM(3)   INFORM(4)   INFORM(5)
         0           0           0           0           0

 ICM ===>  SMOOTH CROSS SECTION

    LAST    NOARY  NDATA(1) NDATA(2) NDATA(3) NDATA(4)
```

(Continued)

```
      12          3          10          1          1

 ICM ===>  SMOOTH CROSS SECTION

    LAST     NOARY  NDATA(1) NDATA(2) NDATA(3) NDATA(4)
    1370         1     1370

   NTH  NADAT  NDASET
    4     18     15

   DATE     NODE(1)  NODE(2)  NODE(3)   NODE(4)   NODE(5)   NODE(6)   NODE(7)   NODE(8)   NODE(9)   NODE(10)
 XX-XX-XX     J137     INFX     1261       ELA

  INFORM(1)   INFORM(2)   INFORM(3)   INFORM(4)   INFORM(5)
          0           0           0           0           0

 ICM ===>  ELASTIC SCATTERING MATRIX

    LAST     NOARY  NDATA(1) NDATA(2) NDATA(3) NDATA(4)
       0         0        0

 ICM ===>  ELASTIC SCATTERING MATRIX

    LAST     NOARY  NDATA(1) NDATA(2) NDATA(3) NDATA(4)
       0         0        0

 ICM ===>  ELASTIC SCATTERING MATRIX

    LAST     NOARY  NDATA(1) NDATA(2) NDATA(3) NDATA(4)
       0         0        0

 ICM ===>  ELASTIC SCATTERING MATRIX

    LAST     NOARY  NDATA(1) NDATA(2) NDATA(3) NDATA(4)
       0         0        0

 ICM ===>  ELASTIC SCATTERING MATRIX

    LAST     NOARY  NDATA(1) NDATA(2) NDATA(3) NDATA(4)
       0         0        0

 ICM ===>  ELASTIC SCATTERING MATRIX

    LAST     NOARY  NDATA(1) NDATA(2) NDATA(3) NDATA(4)
       0         0        0

 ICM ===>  ELASTIC SCATTERING MATRIX

    LAST     NOARY  NDATA(1) NDATA(2) NDATA(3) NDATA(4)
       0         0        0

 ICM ===>  ELASTIC SCATTERING MATRIX

    LAST     NOARY  NDATA(1) NDATA(2) NDATA(3) NDATA(4)
       0         0        0

 ICM ===>  ELASTIC SCATTERING MATRIX

    LAST     NOARY  NDATA(1) NDATA(2) NDATA(3) NDATA(4)
       0         0        0

 ICM ===>  ELASTIC SCATTERING MATRIX
```

7.8 JOB CONTROL CARDS OF MAIL-DAR CODE

The job control cards of linkage and a sample execution of "MAIL-DAR" code are shown below. The index references and the logical unit table are given in page 63, also a sample output is shown in page 64 to 65.

```
INDEX NO.
---------
          //JCLG JOB
          // EXEC JCLG
          //SYSIN DD DATA,DLM='++'
          ************************************************************
          *         USER ACCOUNT INFORMATION CARDS ARE OMITTED       *
          ************************************************************
          //*************************************** START OF LINKAGE
          //LINK  EXEC PGM=JQAL,REGION=768K,COND=(4,LT),
          //        PARM='NOMAP,LET,LIST'
      1   //SYSLIB        DD   DSN=SYS1.FORTLIB,DISP=SHR
      2   //              DD   DSN=SYS9.SSL.LOAD(EXPG2S),DISP=SHR
          //SYSPRINT      DD   SYSOUT=*,DCB=(BLKSIZE=4840)
          //SYSTERM       DD   SYSOUT=*
          //SYSUT1        DD   UNIT=VIO,SPACE=(TRK,(30,10))
      3   //SYSLMOD       DD   DSN=J????.MAILDAR.LOAD,UNIT=TSSWK,
          //           DISP=(NEW,CATLG,DELETE),SPACE=(TRK,(30,10,1),RLSE),
          //              DCB=(BLKSIZE=19069,RECFM=U)
      4   //SYSLIN        DD   DSN=J????.MAILDAR.OBJ,DISP=SHR
      5   //              DD   DSN=J????.GETDCB.OBJ,DISP=SHR
      6   //              DD   DSN=J????.ALLOC.OBJ,DISP=SHR
          //*************************************** END OF LINKAGE
          //*************************************** START OF EXECUTION
          //RUN   EXEC PGM=TEMPNAME,COND=(4,LT),PARM='FLIB(ERRCUT=0)'
      7   //STEPLIB       DD  DSN=J????.MAILDAR.LOAD,DISP=SHR
          //SYSPRINT      DD  SYSOUT=*,
          //              DCB=(RECFM=FBA,LRECL=137,BLKSIZE=19043)
          //FT05F001      DD DDNAME=SYSIN
          //FT06F001      DD  SYSOUT=*,
          //              DCB=(RECFM=FBA,LRECL=137,BLKSIZE=19043)
      8   //FT01F001 DD DSN=J????.CL137300.DATA,DISP=SHR
      9   //FT02F001 DD DSN=FT02.DATA,DISP=(NEW,PASS,DELETE),
          //         UNIT=TSSWK,SPACE=(TRK,(30,30)),
          //         DCB=(LRECL=6208,BLKSIZE=6212,RECFM=VBS,DSORG=PS)
     10   //FT03F001 DD DSN=J????.MACRO1.DATA,DISP=(NEW,CATLG,DELETE),
          //         UNIT=TSSWK,SPACE=(TRK,(300,30)),
          //         DCB=(LRECL=6208,BLKSIZE=6212,RECFM=VBS,DSORG=PS)
     11   //FT11F001 DD DSN=FT11.DATA,DISP=(NEW,PASS,DELETE),
          //         UNIT=TSSWK,SPACE=(TRK,(30,30)),
          //         DCB=(LRECL=6208,BLKSIZE=6212,RECFM=VBS,DSORG=PS)
     12   //FT15F001 DD DSN=FT15.DATA,DISP=(NEW,PASS,DELETE),
          //         UNIT=TSSWK,SPACE=(TRK,(30,30)),
          //         DCB=(LRECL=6208,BLKSIZE=6212,RECFM=VBS,DSORG=PS)
     13   //FT31F001 DD DSN=J????.SM137300.DATA,DISP=SHR
     14   //FT32F001 DD DSN=FT32.DATA,DISP=(NEW,PASS,DELETE),
          //         UNIT=TSSWK,SPACE=(TRK,(30,30)),
          //         DCB=(LRECL=6208,BLKSIZE=6212,RECFM=VBS,DSORG=PS),
     15   //FT33F001 DD DSN=FT33.DATA,DISP=(NEW,PASS,DELETE),
          //         UNIT=TSSWK,SPACE=(TRK,(30,30)),
          //         DCB=(LRECL=6208,BLKSIZE=6212,RECFM=VBS,DSORG=PS)
     16   //FT91F001 DD DSN=J????.DPS.DATA,DISP=SHR
          //SYSIN DD *
     17       1  137    1    0
              3   64
             92    4  114    4  137    4
          ---- ( U235 + U238 + U234 + U236 ) ----   CUMLIB DATA SET
              4    0    0   10 0.0001
               4922350     4922380     4922340     4922360
           4.48006E-02 2.65780E-03 4.82700E-04 9.57000E-05
          ++
          //
```

INDEX REFERENCES

1 : FORTRAN-77 LIBRARY.
2 : FACOM SCIENTIFIC SUBROUTINE.
3 : The created load module of MAIL-DAR CODE.
4 : The object module which the fortran source program compiled.
5 : The object module which the assembler program GETDCB assembled.
6 : The object module which the assembler program ALLOC assembled.
7 : The load module as samc as index 2.
8 : MGCL LIBRARY.
9 : Temporary file, VBS file required.
10 : The created macro-scopic cross section.
11 : Temporary file, VBS file required.
12 : Temporary file, VBS file required.
13 : SMF LIBRARY.
14 : Temporary file, VBS file required.
15 : Temporary file, VBS file required.
16 : The data pool which the micro-scopic cross section stored.
17 : Input data.

Table 7.8.1 LOGICAL UNIT TABLE OF MAIL-DAR CODE

LOGICAL UNITS	SPECIFICATIONS
FT01F001	Allocate to the MGCL LIBRARY.
FT02F001	Temporary file, VBS file required.
FT03F001	Output file of the macroscopic cross section, VBS file required.
FT05F001	INPUT DATA READ.
FT06F001	OUTPUT DATA WRITE.
FT11F001	Temporary file, VBS file required.
FT15F001	Temporary file, VBS file required.
FT31F001	Allocate to the SMF LIBRARY.
FT32F001	Temporary file, VBS file required.
FT33F001	Temporary file, VBS file required.
FT91F001	Allocate to the data pool which contains the microscopic cross section data (= Double differential cross section).

A sample output of REMAIL-DAR code

The first couple of pages of the print out of the "REMAIL-DAR" code are shown below.

```
                                                INPUT DATA IMAGE LIST

                CARD          1         2         3         4         5         6         7         8
                 SEQ.----5----0----5----0----5----0----5----0----5----0----5----0----5----0----5----0
                   1     1  137    1    0
                   2     3   64
                   3    92     4  114     4  137     4
                   4 ---- ( U235 + U238 + U234 + U236 ) ----   CUMLIB DATA SET
                   5     4     0    0   10 0.0001
                   6       4922350      4922380      4922340      4922360
                   7  4.48006E-02 2.65780E-03 4.82700E-04 9.57000E-05
                     ----5----0----5----0----5----0----5----0----5----0----5----0----5----0----5----0

 CUMULATIVE DATA LENGTH =      41772

 NO. OF ANGLE
       4          4          4          4          4          4          4          4          4          4
       4          4          4          4          4          4          4          4          4          4
       4          4          4          4          4          4          4          4          4          4
       4          4          4          4          4          4          4          4          4          4
       4          4          4          4          4          4          4          4          4          4
       4          4          4          4          4          4          4          4          4          4
       4          4          4          4          4          4          4          4          4          4
       4          4          4          4          4          4          4          4          4          4
       4          4          4          4          4          4          4          4          4          4
       4          4          4          4          4          4          4          4          4          4
       4          4          4          4          4          4          4          4          4          4
       4          4          4          4          4          4          4          4          4          4
       4          4          4          4          4          4          4          4          4          4
       4          4          4          4          4          4          4

 ***** DYNAMIC ALLOCATION RESULT *****
   =====>     166442 WORD ALLOCATED

 ***** DYNAMIC ALLOCATION RESULT *****
   =====>     665768 BYTE ALLOCATED


 ******************************  GENERAL INFORMATION  ******************************

   ** NO. OF REGION    ---     1
   ** NO. OF GROUP     ---   137
   ** PRINT OPTION     ---     1
   ** SPECTRUM OPTION ---     0
      IF .EQ. 1 : U -235
      IF .EQ. 2 : PU-239
```

(Continued)

```
REGION ---     1

          * TITLE ---  ---- ( U235 + U238 + U234 + U236 ) ----   CUMLIB DATA SET

          * NMAX   ---     4

          * IREG   ---     0

          * IPRINT ---     0

          * NSTOP  ---    10

          * EPSL   ---    1.0000E-04

 USED AREA ---      65705   WITHIN ---     166442

   **MAT-NO.**          **DENSITY**

      4922350            4.4801E-02
      4922380            2.6578E-03
      4922340            4.8270E-04
      4922360            9.5700E-05

  IREG=     0
```

4922350 U-235

GROUP	ABSORPTION	NYU-FISSION	TOTAL	ELASTIC	IN-ELASTIC	N-2N	FISSION
1	2.2301E-00	1.0235E+01	5.8929E+00	3.2677E+00	2.1943E-01	1.7535E-01	2.2281E+00
2	2.0967E-00	9.1139E+00	5.7965E+00	3.1503E+00	2.2030E-01	3.2894E-01	2.0942E+00
3	1.7976E+00	7.4231E+00	5.7401E+00	3.3001E+00	1.9346E-01	4.4885E-01	1.7947E+00
4	1.7401E-00	6.8465E+00	5.7672E+00	3.3989E+00	1.7647E-01	4.5177E-01	1.7362E+00
5	1.7971E-00	6.7636E+00	5.9056E+00	3.4858E+00	1.8921E-01	4.3369E-01	1.7913E+00
6	1.7871E-00	6.4533E+00	6.1458E+00	3.7004E+00	2.7209E-01	3.8687E-01	1.7786E+00
7	1.6439E-00	5.6854E+00	6.4876E+00	4.0008E+00	5.3483E-01	3.1016E-01	1.6325E+00
8	1.3545E-00	4.4805E+00	6.8925E+00	4.2733E+00	1.0834E+00	1.8413E-01	1.3400E+00
9	1.1048E-00	3.5045E+00	7.2819E+00	4.4746E+00	1.6599E+00	4.3548E-02	1.0886E+00
10	1.1114E+00	3.3960E+00	7.6082E+00	4.6853E+00	1.8118E+00	1.8658E-05	1.0921E+00
11	1.1457E+00	3.3856E+00	7.8376E+00	4.8632E+00	1.8287E+00	0.0	1.1232E+00
12	1.1819E+00	3.3866E+00	7.9365E+00	4.9365E+00	1.8181E+00	0.0	1.1559E+00
13	1.2181E+00	3.3919E+00	7.9179E+00	4.8895E+00	1.8102E+00	0.0	1.1883E+00
14	1.2612E+00	3.4214E+00	7.8235E+00	4.7555E+00	1.8068E+00	0.0	1.2266E+00
15	1.2999E+00	3.4591E+00	7.6730E+00	4.6001E+00	1.7730E+00	0.0	1.2605E+00
16	1.3231E+00	3.4604E+00	7.4760E+00	4.4027E+00	1.7501E+00	0.0	1.2780E+00
17	1.3274E+00	3.4148E+00	7.2534E+00	4.1766E+00	1.7494E+00	0.0	1.2764E+00
18	1.3273E+00	3.3590E+00	7.0421E+00	4.0098E+00	1.7052E+00	0.0	1.2691E+00
19	1.3259E+00	3.3007E+00	6.8732E+00	3.9073E+00	1.6401E+00	0.0	1.2590E+00
20	1.3283E+00	3.2498E+00	6.7622E+00	3.8612E+00	1.5728E+00	0.0	1.2502E+00
21	1.3467E+00	3.2388E+00	6.7168E+00	3.8747E+00	1.4954E+00	0.0	1.2555E+00
22	1.3569E-00	3.2056E+00	6.7476E+00	3.9639E+00	1.4267E+00	0.0	1.2514E+00
23	1.3361E+00	3.1006E+00	6.8569E+00	4.1563E+00	1.3645E+00	0.0	1.2184E+00
24	1.2884E+00	2.9434E+00	7.0317E+00	4.4033E+00	1.3399E+00	0.0	1.1634E+00
25	1.2637E+00	2.8464E+00	7.2465E+00	4.6552E+00	1.3276E+00	0.0	1.1309E+00
26	1.2816E+00	2.8513E+00	7.4982E+00	4.9394E+00	1.2770E+00	0.0	1.1381E+00
27	1.3076E+00	2.8726E+00	7.7895E+00	5.3248E+00	1.1569E+00	0.0	1.1513E+00
28	1.3312E+00	2.8893E+00	8.1053E+00	5.6646E+00	1.1095E+00	0.0	1.1621E+00
29	1.3697E+00	2.9397E+00	8.3808E+00	5.9464E+00	1.0644E+00	0.0	1.1862E+00
30	1.4089E+00	2.9938E+00	8.6563E+00	6.2709E+00	9.7626E-01	0.0	1.2114E+00
31	1.4318E+00	3.0155E+00	8.9300E+00	6.5940E+00	9.0406E-01	0.0	1.2233E+00
32	1.4710E+00	3.0649E+00	9.2032E+00	6.8892E+00	8.4285E-01	0.0	1.2461E+00
33	1.5432E+00	3.1698E+00	9.4689E+00	7.1405E+00	7.8499E-01	0.0	1.2912E+00
34	1.5805E+00	3.2026E+00	9.7429E+00	7.4414E+00	7.2092E-01	0.0	1.3068E+00
35	1.6095E+00	3.2340E+00	1.0020E+01	7.7534E+00	6.5667E-01	0.0	1.3217E+00
36	1.6664E+00	3.3251E+00	1.0293E+01	8.0104E+00	6.1607E-01	0.0	1.3607E+00
37	1.7388E+00	3.4417E+00	1.0565E+01	8.2681E+00	5.5806E-01	0.0	1.4101E+00
38	1.8085E+00	3.5475E+00	1.0835E+01	8.5206E+00	5.0566E-01	0.0	1.4550E+00
39	1.8764E+00	3.6468E+00	1.1105E+01	8.7651E+00	4.6290E-01	0.0	1.4971E+00
40	1.9394E+00	3.7366E+00	1.1379E+01	9.0135E+00	4.2600E-01	0.0	1.5353E+00
41	2.0067E+00	3.8366E+00	1.1653E+01	9.2573E+00	3.8855E-01	0.0	1.5775E+00
42	2.0345E+00	3.8627E+00	1.1921E+01	9.6019E+00	2.8455E-01	0.0	1.5892E+00
43	2.1415E+00	4.0317E+00	1.2190E+01	9.8635E+00	1.8484E-01	0.0	1.6597E+00
44	2.3112E+00	4.3160E+00	1.2439E+01	9.9880E+00	1.3934E-01	0.0	1.7776E+00
45	2.3566E+00	4.3707E+00	1.2674E+01	1.0217E+01	1.0088E-01	0.0	1.8010E+00
46	2.4826E+00	4.5734E+00	1.2906E+01	1.0357E+01	6.6933E-02	0.0	1.8852E+00
47	2.3867E+00	4.3646E+00	1.3113E+01	1.0686E+01	4.0944E-02	0.0	1.7998E+00
48	2.5255E+00	4.5822E+00	1.3316E+01	1.0757E+01	3.3005E-02	0.0	1.8901E+00
49	2.5691E+00	4.6256E+00	1.3510E+01	1.0913E+01	2.7465E-02	0.0	1.9085E+00
50	2.7351E+00	4.8915E+00	1.3694E+01	1.0937E+01	2.2577E-02	0.0	2.0187E+00
51	2.8788E+00	5.1240E+00	1.3869E+01	1.0972E+01	1.8263E-02	0.0	2.1151E+00
52	2.8901E+00	5.1282E+00	1.4033E+01	1.1129E+01	1.4456E-02	0.0	2.1172E+00
53	3.1069E+00	5.4835E+00	1.4316E+01	1.1198E+01	1.1096E-02	0.0	2.2643E+00
54	3.0650E+00	5.2932E+00	1.4348E+01	1.1274E+01	8.1311E-03	0.0	2.1860E+00
55	3.2487E+00	5.7003E+00	1.4567E+01	1.1313E+01	5.5147E-03	0.0	2.3544E+00
56	3.2995E+00	5.6868E+00	1.4689E+01	1.1386E+01	3.2058E-03	0.0	2.3491E+00
57	3.6542E+00	6.3681E+00	1.5130E+01	1.1475E+01	5.9394E-04	0.0	2.6309E+00

7.9 JOB CONTROL CARDS OF REMAIL-DAR CODE

The job control cards of the linkage and a sample execution of "REMAIL-DAR" are shown below. The index references and logical unit table are shown in page 67. A sample output is given in page 68.

```
INDEX NO.
---------
          //JCLG JOB
          // EXEC JCLG
          //SYSIN DD DATA,DLM='++'
          ************************************************************
          *        USER ACCOUNT INFORMATION CARDS ARE OMITTED         *
          ************************************************************
          //*********************************************** START OF LINKAGE
          //LINK  EXEC PGM=JQAL,REGION=2048K,COND=(4,LT),
          //      PARM='NOMAP,LET,LIST'
     1    //SYSLIB       DD   DSN=SYS1.FORTLIB,DISP=SHR
          //SYSPRINT     DD   SYSOUT=*,DCB=(BLKSIZE=4840)
          //SYSTERM      DD   SYSOUT=*
          //SYSUT1       DD   UNIT=VIO,SPACE=(TRK,(30,10))
     2    //SYSLMOD      DD   DSN=J????.REMAIL.LOAD,UNIT=TSSWK,
          //             DISP=(NEW,CATLG,DELETE),SPACE=(TRK,(30,10,1),RLSE),
          //             DCB=(BLKSIZE=19069,RECFM=U)
     3    //SYSLIN       DD   DSN=J????.REMAIL.OBJ,DISP=SHR
          //************************************************* END OF LINKAGE
          //************************************************ START OF EXECUTION
          //RUN   EXEC PGM=TEMPNAME,COND=(4,LT),PARM='FLIB(ERRCUT=0)'
     4    //STEPLIB      DD  DSN=J????.REMAIL.LOAD,DISP=SHR
          //SYSPRINT     DD  SYSOUT=*,
          //             DCB=(RECFM=FBA,LRECL=137,BLKSIZE=19043)
          //FT05F001     DD DDNAME=SYSIN
          //FT06F001     DD  SYSOUT=*,
          //             DCB=(RECFM=FBA,LRECL=137,BLKSIZE=19043)
     5    //FT10F001 DD DSN=J????.MACRO1.DATA,DISP=SHR
     6    //FT11F001 DD DSN=J????.MACRO2.DATA,DISP=SHR
     7    //FT50F001 DD DSN=J????.NEWMACRO.DATA,DISP=(NEW,CATLG,DELETE),
          //         UNIT=D0950,SPACE=(TRK,(100,10)),
          //         DCB=(LRECL=6208,BLKSIZE=6208,RECFM=VBS,DSORG=PS)
          //SYSIN DD *
     8      137    2    2    0
              1    1
              1
              2
          ++
          //
```

INDEX REFERENCES

1 : FORTRAN-77 LIBRARY.

2 : The created load module of the REMAIL-DAR CODE.

3 : The object module which the fortran source program compiled.

4 : The load module as same as INDEX 2.

5 : The macroscopic cross section data created by MAIL-DAR code.

6 : The macroscopic cross section data created by MAIL-DAR code, however, the different material to the INDEX 5.

7 : The created new macroscopic cross section by combinating INDEX 5 and INDEX 6.

8 : Input data.

Table 7.9.1 LOGICAL UNIT TABLE OF REMAIL-DAR CODE

LOGICAL UNITS	SPECIFICATIONS
FT05F001	INPUT DATA READ.
FT06F001	OUTPUT DATA WRITE.
FT10F001	Allocate to the macroscopic cross section files.
⎫ ⎬ FT49F001	Allocation should be started from FT10F001.
FT50F001	Output file of the new macroscopic cross section, VBS file required.

A sample output of REMAIL-DAR code

The first couple of pages of the print out of the "REMAIL-DAR" code are shown below.

```
        *** LIBRARY PROCESSING PROGRAM FOR KENO-DAR ONLY

 GROUP  --- 137

 REGION ---   2

 FILE   ---   2

 IPRINT ---   2   (IF =1  PRINT OUT NEW LIBRARY)


FILE  REGION  NEW REGION ID. (0=SKIP)
UNIT  NO.        1    2    3    4    5    6    7    8    9   10   11   12   13   14   15   16   17   18   19   20

  10     1        1

  11     1        2


NUMBER OF ANGULAR MESHES ---  64

*** REGION ID OF REARRANGED KENO-DAR LIBRARY ***

REGION  1 = ---- ( U235 + U238 + U234 + U236 ) ----   CUMLIB DATA SET

REGION  2 = ---- ( H1 + C12 ) ------   CUMLIB DATA SET



**** KENO-DAR LIBRARY PRINT OUT ****


         2       137       137


1.64870E+07 1.45500E+07 1.28400E+07 1.13310E+07 1.00000E+07 8.82500E+06 7.78800E+06 6.87290E+06 6.06530E+06 5.35260E+06
4.72370E+06 4.16860E+06 3.67880E+06 3.24650E+06 2.86500E+06 2.52840E+06 2.23130E+06 1.96910E+06 1.73770E+06 1.53350E+06
1.35340E+06 1.19430E+06 1.05400E+06 9.30140E+05 8.20850E+05 7.24400E+05 6.39280E+05 5.64160E+05 4.97870E+05 4.39370E+05
3.87740E+05 3.42180E+05 3.01970E+05 2.66490E+05 2.35180E+05 2.07540E+05 1.83160E+05 1.61630E+05 1.42640E+05 1.25880E+05
1.11070E+05 9.80370E+04 8.65170E+04 7.63510E+04 6.73790E+04 5.94620E+04 5.24750E+04 4.63090E+04 4.08680E+04 3.60660E+04
3.18280E+04 2.80880E+04 2.47880E+04 2.18750E+04 1.93050E+04 1.70360E+04 1.50340E+04 1.17090E+04 9.11880E+03 7.10170E+03
5.53050E+03 4.30740E+03 3.35460E+03 2.61260E+03 2.03470E+03 1.58460E+03 1.23410E+03 9.61120E+02 7.48520E+02 5.82950E+02
4.54000E+02 3.53580E+02 2.75360E+02 2.14450E+02 1.67020E+02 1.30070E+02 1.01300E+02 7.88930E+01 6.14420E+01 4.78510E+01
3.72670E+01 2.90230E+01 2.26030E+01 1.76030E+01 1.37100E+01 1.06770E+01 8.31530E+00 6.47600E+00 5.04350E+00 3.92790E+00
3.05900E+00 2.38240E+00 1.85540E+00 1.63740E+00 1.44500E+00 1.27520E+00 1.12540E+00 9.93120E-01 8.76420E-01 7.73440E-01
6.82560E-01 6.02360E-01 5.31580E-01 4.69120E-01 4.13990E-01 3.89250E-01 3.65280E-01 3.42060E-01 3.19610E-01 2.97920E-01
2.76990E-01 2.56830E-01 2.37420E-01 2.18780E-01 2.00900E-01 1.83780E-01 1.67430E-01 1.51830E-01 1.37000E-01 1.22930E-01
1.09620E-01 9.70780E-02 8.52950E-02 7.42740E-02 6.40150E-02 5.45180E-02 4.57830E-02 3.78110E-02 3.06000E-02 2.41520E-02
1.84650E-02 1.35410E-02 9.37920E-03 5.97930E-03 3.34140E-03 1.46570E-03 3.52090E-04 3.30660E-05
4.99957E-01-3.75006E-01-2.49980E-01-1.24957E-01 0.0         1.24996E-01 2.50001E-01 3.74999E-01 5.00001E-01 6.25003E-01
7.49993E-01 8.75005E-01 9.99998E-01 1.12501E+00 1.25002E+00 1.37500E+00 1.50000E+00 1.62501E+00 1.75002E+00 1.87503E+00
1.99996E+00 2.12502E+00 2.24999E+00 2.37500E+00 2.50000E+00 2.62500E+00 2.75000E+00 2.87500E+00 3.00000E+00 3.12500E+00
3.25000E+00 3.37500E+00 3.50001E+00 3.62500E+00 3.74999E+00 3.87502E+00 3.99998E+00 4.12503E+00 4.25002E+00 4.37501E+00
4.50000E+00 4.62500E+00 4.75000E+00 4.87500E+00 5.00001E+00 5.12500E+00 5.25000E+00 5.37500E+00 5.49999E+00 5.62499E+00
5.74999E+00 5.87500E+00 5.99998E+00 6.12500E+00 6.24998E+00 6.37501E+00 6.50003E+00 6.74998E+00 7.00000E+00 7.25001E+00
7.50001E+00 7.75001E+00 8.00001E+00 8.24999E+00 8.49999E+00 8.75001E+00 9.00000E+00 9.25000E+00 9.50000E+00 9.74999E+00
1.00000E+01 1.02500E+01 1.05000E+01 1.07500E+01 1.10000E+01 1.12500E+01 1.15000E+01 1.17500E+01 1.20000E+01 1.22500E+01
1.25000E+01 1.27500E+01 1.30000E+01 1.32500E+01 1.35000E+01 1.37500E+01 1.40000E+01 1.42500E+01 1.45000E+01 1.47500E+01
1.50000E+01 1.52500E+01 1.55000E+01 1.56250E+01 1.57500E+01 1.58750E+01 1.60000E+01 1.61250E+01 1.62500E+01 1.63750E+01
1.65000E+01 1.66250E+01 1.67500E+01 1.68750E+01 1.70000E+01 1.70616E+01 1.71252E+01 1.71909E+01 1.72587E+01 1.73290E+01
1.74019E+01 1.74774E+01 1.75560E+01 1.76378E+01 1.77230E+01 1.78121E+01 1.79053E+01 1.80031E+01 1.81059E+01 1.82142E+01
1.83285E+01 1.84503E+01 1.85797E+01 1.87181E+01 1.88667E+01 1.90273E+01 1.92019E+01 1.93932E+01 1.96048E+01 1.98415E+01
2.01100E+01 2.04201E+01 2.07874E+01 2.12375E+01 2.18195E+01 2.26435E+01 2.40697E+01 2.64351E+01
```

7.10 JOB CONTROL CARDS OF MKENO-DAR CODE

The job control cards of the compilation of a subroutine "ALOCAT" which gives the dimension size throughout the "MKENO-DAR" code and the linkage and a sample execution are shown below. The index references and the logical unit table are shown in page 71 to 72. A sample output is shown in page 73 to 75.

```
INDEX NO.
---------
          //JCLG JOB
          // EXEC JCLG
          //SYSIN DD DATA,DLM='++'
          ************************************************************
          *         USER ACCOUNT INFORMATION CARDS ARE OMITTED        *
          ************************************************************
          //*********************************************** START OF COMPILATION 1
          //FORT     EXEC PGM=JZKAFORT,REGION=4096K,COND=(4,LT),
          //         PARM='OPTIMIZE(2),LINECOUNT(0),NOS,NOSOURCE,NOMAP'
          //SYSPRINT  DD SYSOUT=*,
          //            DCB=(RECFM=FBA,LRECL=137,BLKSIZE=19043)
          //SYSTERM   DD SYSOUT=*,
          //            DCB=(RECFM=FBA,LRECL=137,BLKSIZE=19043)
      1   //SYSLIN    DD DSN=J????.ALOCAT.OBJ,DISP=(NEW,CATLG,DELETE),
          //            UNIT=TSSWK,SPACE=(TRK,(30,10)),DCB=BLKSIZE=3200
          //SYSIN     DD *
      2             SUBROUTINE ALOCAT( PROGM )
                    COMMON D( 450000 )
                    CALL PROGM( D , 450000 )
                    RETURN
                    END
          //*********************************************** END OF COMPILATION 1
          //*********************************************** START OF LINKAGE
          //LINK  EXEC PGM=JQAL,REGION=768K,COND=(4,LT),
          //         PARM='NOMAP,LIST,LET'
      3   //SYSLIB       DD   DSN=SYS1.FORTLIB,DISP=SHR
      4   //               DD   DSN=SYS9.NO.LOAD,DISP=SHR
      5   //               DD   DSN=SYS9.GDS.LOAD,DISP=SHR
      6   //               DD   DSN=SYS9.GGS.LOAD,DISP=SHR
          //SYSPRINT     DD   SYSOUT=*,DCB=(BLKSIZE=4840)
          //SYSTERM      DD   SYSOUT=*
          //SYSUT1       DD   UNIT=VIO,SPACE=(TRK,(30,10))
      7   //SYSLMOD      DD   DSN=J????.MKENO.LOAD,UNIT=TSSWK,
          //          DISP=(NEW,CATLG,DELETE),SPACE=(TRK,(30,10,1),RLSE),
          //               DCB=(BLKSIZE=19069,RECFM=U)
      8   //SYSLIN       DD   DSN=J????.MKENO.OBJ,DISP=SHR
      9   //               DD   DSN=J????.ALOCAT.OBJ,DISP=SHR
          //*********************************************** END OF LINKAGE
```

INDEX NO. (CONTINUED)

```
//********************************************* START OF EXECUTION
//RUN   EXEC PGM=TEMPNAME,COND=(4,LT),PARM='FLIB(ERRCUT=0)'
10    //STEPLIB      DD  DSN=J????.MKENO.LOAD,DISP=SHR
      //SYSPRINT     DD  SYSOUT=*,
      //             DCB=(RECFM=FBA,LRECL=137,BLKSIZE=19043)
      //FT05F001     DD DDNAME=SYSIN
      //FT06F001     DD  SYSOUT=*,
      //             DCB=(RECFM=FBA,LRECL=137,BLKSIZE=19043)
11    //FT10F001 DD DSN=FT10.DATA,DISP=(NEW,PASS,DELETE),
      //        UNIT=TSSWK,SPACE=(TRK,(30,30)),
      //  DCB=(LRECL=6208,BLKSIZE=6212,RECFM=VBS,DSORG=PS)
12    //FT18F001 DD DSN=FT18.DATA,DISP=(NEW,PASS,DELETE),
      //        UNIT=TSSWK,SPACE=(TRK,(30,30)),
      //  DCB=(LRECL=6208,BLKSIZE=6212,RECFM=VBS,DSORG=PS)
13    //FT41F001 DD DSN=J????.NEWMACRO.DATA,DISP=SHR
14    //FT42F001 DD DSN=FT42.DATA,DISP=(NEW,PASS,DELETE),
      //        UNIT=TSSWK,SPACE=(TRK,(30,30)),
      //  DCB=(LRECL=6208,BLKSIZE=6212,RECFM=VBS,DSORG=PS)
15    //FT44F001 DD DSN=FT44.DATA,DISP=(NEW,PASS,DELETE),
      //        UNIT=TSSWK,SPACE=(TRK,(30,30)),
      //  DCB=(LRECL=6208,BLKSIZE=6212,RECFM=VBS,DSORG=PS)
16    //FT45F001 DD DSN=FT45.DATA,DISP=(NEW,PASS,DELETE),
      //        UNIT=TSSWK,SPACE=(TRK,(30,30)),
      //  DCB=(LRECL=6208,BLKSIZE=6212,RECFM=VBS,DSORG=PS)
17    //FT50F001 DD DSN=FT50.DATA,DISP=(NEW,PASS,DELETE),
      //        UNIT=TSSWK,SPACE=(TRK,(30,30)),
      //  DCB=(LRECL=6208,BLKSIZE=6212,RECFM=VBS,DSORG=PS)
18    //FT51F001 DD DSN=FT51.DATA,DISP=(NEW,PASS,DELETE),
      //        UNIT=TSSWK,SPACE=(TRK,(30,30)),
      //  DCB=(LRECL=80,BLKSIZE=3200,RECFM=FB,DSORG=PS)
      //SYSIN DD *
19     DAR.MANUAL SANPLE RUN DATA
       59.0   23   20  3  137 137  1  1  1  6  1  1  1  1  1  1  0
       2000   10RO  1
      6R-1.0
       1  -1   1.0
      SUPER BOX   1  1  1  1  1
      BOX TYPE  1
      CYLINDER  1  10.0    5.0  -5.0                      137RO.5
      CUBOID    0  40.0  -40.0  40.0  -40.0  5.0  -5.0  137RO.5
      CELL BDY  0  40.0  -40.0  40.0  -40.0  5.0  -5.0  137RO.5
      CUBOID    0  40.0  -40.0  40.0  -40.0  5.0  -5.0  137RO.5
      CORE BDY  0  40.0  -40.0  40.0  -40.0  5.0  -5.0  137RO.5
      CUBOID    0  40.0  -40.0  40.0  -40.0  5.0  -5.0  137RO.5
      -1
      END KENO
      /*
      ++
      //
```

INDEX REFERENCES

1 : The object module which obtained by the compilation of the given subroutine.

2 : The source subroutine. If the memory size of the MKENO-DAR code is short, give the greater value to the dimension D.

3 : FORTRAN-77 LIBRARY.

4 : GRAPHIC LIBRARY.

5 : GRAPHIC LIBRARY.

6 : GRAPHIC LIBRARY.

7 : The created load module of MKENO-DAR code.

8 : The object module which is obtained by the compilation of the source program of MKENO-DAR.

9 : The object module as same as INDEX 1.

10 : The load module as same as INDEX 7.

11 : Temporary file, VBS file required.

12 : Temporary file, VBS file required.

13 : The macro-scopic cross section file obtained by REMAIL-DAR code.

14 : Temporary file, VBS file required.

15 : Restart data file. If no restart calculation desired, then the file can be substituted by temporary VBS file or dummy file. If restart calculation desired, then the file should be cataloged with showing format.

16 : Restart data file as same as INDEX 15. If restart calculation desired, the file name should be the same to the INDEX 15, since, INDEX 15 works as a read file, and INDEX 16 works as a write file.

17 : Temporary file, VBS file required.

18 : Temporary file, EBCDIC file required.

19 : Input data.

Table 7.10.1 LOGICAL UNIT TABLE OF MKENO-DAR CODE

LOGICAL UNITS	SPECIFICATIONS
FT04F001 OR NXCUTE	} AMPX WORKING LIBRARY.
FT05F001	INPUT DATA READ.
FT06F001	OUTPUT DATA WRITE.
FT10F001	Temporary file, VBS file required.
FT18F001	Temporary file, VBS file required.
FT41F001 OR NXCUTE	} MACROSCOPIC CROSS SECTION FILE
FT42F001	Temporary file, VBS file required.
FT43F001	Allocate to the WTAVG library for automatic reflector option.
FT44F001	Read restart data for MKENO-DAR.
FT45F001	Write restart data for MKENO-DAR.
FT50F001	Temporary file, VBS file required.
FT51F001	Temporary file, FB file required.
GDFILE	Output file for graphic processing, the unit number of GDFILE is computer system dependent.

A sample output of MKENO-DAR code

The first couple of pages of print out of the "MKENO-DAR" code are shown below.

```
                                    INPUT DATA IMAGE LIST

CARD          1         2         3         4         5         6         7         8
 SEQ.----5----0----5----0----5----0----5----0----5----0----5----0----5----0----5----0
   1 DAR.MANUAL SANPLE RUN DATA
   2  59.0   23   20  3  137 137  1  1  1  6  1  1  1  1  1  1  0
   3  2000   10R0  1
   4  6R-1.0
   5  1  -1   1.0
   6 SUPER BOX   1  1  1  1  1
   7 BOX TYPE  1
   8 CYLINDER  1  10.0    5.0  -5.0                      137R0.5
   9 CUBOID    0  40.0  -40.0  40.0  -40.0  5.0  -5.0  137R0.5
  10 CELL BDY  0  40.0  -40.0  40.0  -40.0  5.0  -5.0  137R0.5
  11 CUBOID    0  40.0  -40.0  40.0  -40.0  5.0  -5.0  137R0.5
  12 CORE BDY  0  40.0  -40.0  40.0  -40.0  5.0  -5.0  137R0.5
  13 CUBOID    0  40.0  -40.0  40.0  -40.0  5.0  -5.0  137R0.5
  14 -1
  15 END KENO
     ----5----0----5----0----5----0----5----0----5----0----5----0----5----0----5----0
```

DAR.MANUAL SANPLE RUN DATA

NUMBER OF GENERATIONS	23	START TYPE	0
NUMBER PER GENERATION	20	GENERATIONS BETWEEN CHECKPOINTS	0
NUMBER OF GENERATIONS TO BE SKIPPED	3	LIST INPUT X-SECTIONS READ FROM TAPE	NO
NUMBER OF ENERGY GROUPS	137	LIST 1-D MIXTURE X SECTIONS	NO
MAX. NUMBER OF ENERGY TRANSFERS	137	LIST 2-D MIXTURE X-SECTIONS	NO
NUMBER OF INPUT NUCLIDES	1	BALANCE TABLE TYPE	0
NUMBER OF MIXTURES	1	USE X-SECTIONS FROM PREVIOUS CASE	NO
NUMBER OF MIXING TABLE ENTRIES	1	USE GEOMETRY FROM PREVIOUS CASE	NO
NUMBER OF GEOMETRY CARDS	6	USE VELOCITIES FROM PREVIOUS CASE	NO
NUMBER OF BOX TYPES	1	COMPUTE MATRIX K-EFFECTIVE BY UNIT	NO
NUMBER OF UNITS IN X DIRECTION	1	COMPUTE MATRIX K-EFFECTIVE BY BOX TYPE	NO
NUMBER OF UNITS IN Y DIRECTION	1	LIST FISS PROB MATRIX BY UNIT	NO
NUMBER OF UNITS IN Z DIRECTION	1	ADJOINT CALCULATION	NO
NUMBER OF NUCLIDES READ FROM TAPE	1	USE EXPONENTIAL TRANSFORM	NO
ALBEDO TYPE	1	O/N = K-EFFCTIVE/FIXED SOURCE PROB.	0
SEARCH TYPE	0	CALCULATE FISSION DENSITIES	YES
SCATTERING MATRIX 0/1 = P-1/CUMULATIVE	1		

```
THIS PROBLEM WILL BE RUN WITH SPECULARLY REFLECTING BOUNDARY CONDITION

THE ALBEDOS ARE +X = 1.00000E+00 -X = 1.00000E+00 +Y = 1.00000E+00 -Y = 1.00000E+00 +Z = 1.00000E+00 -Z = 1.00000E+00

MAXIMUM TIME =  59.0000 MINUTES

    CUMULATIVE TABLE LENGTH =      41772
    ANGLE MESH NO.          =         64
    FAST AND THERMAL GROUP  =   92   45

    MNANG,MITB,MANG,MMU,MFIX,MATT=     19592     19729     19866     19931     61703         1

STORAGE LOCATIONS REQUIRED FOR THIS JOB =     110882
REMAINING AVAILABLE LOCATIONS=      20190
```

(Continued)

```
DAR.MANUAL SAMPLE RUN DATA

     MIXTURE  NUCLIDE    DENSITY
        1        -1    1.00000E+00


CROSS SECTIONS READ FROM TAPE
NUCLIDE =          1  ---- ( U235 + U238 + U234 + U236 ) ----   CUMLIB DATA SET



DAR.MANUAL SAMPLE RUN DATA
GEOMETRY DESCRIPTION
REGION

 0   SUPER BOX   1    NBOX =  1    NBXMAX =  1    NBYMAX =  1    NBZMAX =  1
REGION
 1   BOX TYPE    1
REGION
  1  CYLINDER    1  RADIUS = 1.0000E+01   +Z = 5.0000E+00   -Z =-5.0000E+00
  2  CUBOID      0      +X = 4.0000E+01   -X =-4.0000E+01   +Y = 4.0000E+01   -Y =-4.0000E-01   +Z = 5.0000E+00   -Z =-5.0000E+00
CELL BDY.
  1  CELL BDY    0      +X = 4.0000E+01   -X =-4.0000E+01   +Y = 4.0000E+01   -Y =-4.0000E+01   +Z = 5.0000E+00   -Z =-5.0000E+00
  2  CUBOID      0      +X = 4.0000E+01   -X =-4.0000E+01   +Y = 4.0000E+01   -Y =-4.0000E+01   +Z = 5.0000E+00   -Z =-5.0000E+00
REFLECTOR
  1  CORE BDY    0      +X = 4.0000E+01   -X =-4.0000E+01   +Y = 4.0000E+01   -Y =-4.0000E+01   +Z = 5.0000E+00   -Z =-5.0000E+00
  2  CUBOID      0      +X = 4.0000E+01   -X =-4.0000E+01   +Y = 4.0000E+01   -Y =-4.0000E+01   +Z = 5.0000E+00   -Z =-5.0000E+00


      DAR.MANUAL SAMPLE RUN DATA
      WEIGHTING FUNCTION
      BOX TYPE   1                                          GROUP   WTLOW    WTAVG    WT HI
            REGION   1 DEFINED GEOMETRY CARD   1
                                                             1  0.166667  0.500000  1.500000
                                                           GROUPS 1 TO  137 SAME AS ABOVE
            REGION   2 DEFINED GEOMETRY CARD   2
                                                             1  0.166667  0.500000  1.500000
                                                           GROUPS 1 TO  137 SAME AS ABOVE
      CELL BDY.
            REGION   1 DEFINED GEOMETRY CARD   3
                                                             1  0.166667  0.500000  1.500000
                                                           GROUPS 1 TO  137 SAME AS ABOVE
            REGION   2 DEFINED GEOMETRY CARD   4
                                                             1  0.166667  0.500000  1.500000
                                                           GROUPS 1 TO  137 SAME AS ABOVE
      REFLECTOR
            REGION   1 DEFINED GEOMETRY CARD   5
                                                             1  0.166667  0.500000  1.500000
                                                           GROUPS 1 TO  137 SAME AS ABOVE
            REGION   2 DEFINED GEOMETRY CARD   6
                                                             1  0.166667  0.500000  1.500000
                                                           GROUPS 1 TO  137 SAME AS ABOVE
      STORAGE LOCATIONS REQUIRED FOR THIS JOB =    110882
      REMAINING AVAILABLE LOCATIONS=     20178
```

(Continued)

```
DAR.MANUAL SAMPLE RUN DATA

 VOLUMES

 SUPER BOX TYPE      1

 BOX TYPE      1
          REGION DEFINED BY GEOMETRY CARD       1   VOLUME =     3.141591E+03 CM**3        CUMULATIVE VOLUME =     3.141591E+03 CM**3
          REGION DEFINED BY GEOMETRY CARD       2   VOLUME =     6.085841E+04 CM**3        CUMULATIVE VOLUME =     6.400000E+04 CM**3

REFLECTOR VOLUMES -  GEOMETRY CARD       3 IS THE CELL BOUNDARY CARD
          REGION DEFINED BY GEOMETRY CARD       4   VOLUME =     0.0          CM**3        CUMULATIVE VOLUME =     6.400000E+04 CM**3

 TOTAL VOLUMES
    1       3.141591E+03
    2       6.085841E+04

REFLECTOR VOLUMES -  GEOMETRY CARD       5 IS THE CORE BOUNDARY CARD
          REGION DEFINED BY GEOMETRY CARD       6   VOLUME =     0.0          CM**3        CUMULATIVE VOLUME =     6.400000E+04 CM**3

 TOTAL VOLUMES
    1       3.141591E+03
    2       6.085841E+04
    3       0.0
    4       0.0
    5       0.0
    6       0.0

VOLUME FRACTION OF THE CORE CONTAINING FISSILE MATERIAL= 0.490987E-01

START TYPE =     0

 THE NEUTRONS WERE STARTED IN THE ARRAY WITH A FLAT DISTRIBUTION.

   20 NEUTRONS WERE INITIALLY STARTED
0.00050 MINUTES WERE REQUIRED FOR STARTING.
```

```
DAR.MANUAL SAMPLE RUN DATA

  GENERATION   K-EFFECTIVE   ELAPSED TIME(MIN)    AVG. K-EFF       DEVIATION       MATRIX K-EFF
       1       2.40250E+00     4.66666E-03      1.00000E+00      0.0             0.0
       2       2.50243E+00     9.00000E-03      1.00000E+00      0.0             0.0
       3       2.26679E+00     1.20000E-02      2.26679E+00      0.0             0.0
       4       2.16049E+00     1.46667E-02      2.21364E+00      5.31487E-02     0.0
                                                          WARNING - ONLY   19 INDEPENDENT FISSION POINTS WERE GENERATED.
       5       2.02397E+00     1.93333E-02      2.15042E+00      7.02820E-02     0.0
       6       2.49028E+00     2.51667E-02      2.23538E+00      9.84263E-02     0.0
       7       2.49164E+00     2.90000E-02      2.28664E+00      9.18644E-02     0.0
       8       2.38426E+00     3.31667E-02      2.30291E+00      7.67531E-02     0.0
       9       2.32954E+00     3.58333E-02      2.30671E+00      6.49934E-02     0.0
                                                          WARNING - ONLY   16 INDEPENDENT FISSION POINTS WERE GENERATED.
      10       2.26453E+00     4.06667E-02      2.30144E+00      5.65335E-02     0.0
      11       2.31535E+00     4.55000E-02      2.30298E+00      4.98848E-02     0.0
      12       2.21426E+00     5.06667E-02      2.29411E+00      4.54915E-02     0.0
      13       2.64927E+00     5.80000E-02      2.32640E+00      5.23077E-02     0.0
                                                          WARNING - ONLY   17 INDEPENDENT FISSION POINTS WERE GENERATED.
      14       2.39800E+00     6.10000E-02      2.33236E+00      4.81217E-02     0.0
                                                          WARNING - ONLY   16 INDEPENDENT FISSION POINTS WERE GENERATED.
      15       2.28456E+00     6.55000E-02      2.32869E+00      4.44193E-02     0.0
      16       2.36894E+00     7.10000E-02      2.33156E+00      4.12279E-02     0.0
      17       2.41104E+00     7.66667E-02      2.33686E+00      3.87465E-02     0.0
      18       2.33398E+00     8.29999E-02      2.33668E+00      3.62452E-02     0.0
      19       2.15876E+00     9.11666E-02      2.32621E+00      3.56203E-02     0.0
      20       2.17530E+00     9.66666E-02      2.31783E+00      3.46151E-02     0.0
      21       2.29857E+00     1.02667E-01      2.31681E+00      3.27589E-02     0.0
      22       2.25897E+00     1.13667E-01      2.31392E+00      3.12130E-02     0.0
      23       2.28881E+00     1.17667E-01      2.31273E+00      2.97130E-02     0.0

THE MATRIX K-EFF IS THE LARGEST EIGENVALUE OF THE MATRIX OF FISSION PROBABILITIES BY UNIT.
THERE ARE NBXMAX * NBYMAX * NBZMAX UNITS IN AN ARRAY.
```

REFERENCES

1) L.M. PERTIE and N.F. CROSS, "KENO-IV - An Improved Monte Carlo Criticality Program", ORNL 4938 (1975)

2) J. KATAKURA, Y. NAITO and Y. KOMURO, "Development of the Computer Code System JACS for Criticality Safety", Trans. of the ANS VOL.41 (1982)

3) W.W. ENGLE, Jr., "A USERS MANUAL FOR ANISN : A One Dimensional Discrete Ordinates Transport Code with Anisotropic Scattering", K-1693, (1967)

4) Y. NAITOH, Y. TSUNOO, Y. KOMURO, and M. NAKAYAMA, "MKENO-DAR : A Direct Anglur Representation Monte Carlo Code for Criticality Safety Analysis", JAERI-M 84-061 (1984)

5) Y. NAITOH, M. YOKOTA, and K. NAKANO, "MULTI-KENO : A Monte Carlo Code for Criticality Safety Analysis", JAERI-M 83-049 (1983)