KERNFORSCHUNGSZENTRUM

KARLSRUHE

Juni 1973 KFK 1725

Institut für Neutronenphysik und Reaktortechnik

Handling and Service Programs for the Karlsruhe Nuclear Data File KEDAK

Part I: Management and Retrieval Programs

B. Krieg

GESELLSCHAFT
FÜR
KERNFORSCHUNG M.B.H.

KARLSRUHE

Als Manuskript vervielfältigt

Für diesen Bericht behalten wir uns alle Rechte vor

GESELLSCHAFT FÜR KERNFORSCHUNG M. B. H.
KARLSRUHE

KERNFORSCHUNGSZENTRUM KARLSRUHE

KFK 1725

Institut für Neutronenphysik und Reaktortechnik

Handling and service programs for the Karlsruhe Nuclear Data File KEDAK Part I: Management and retrieval programs

B. Krieg

Gesellschaft für Kernforschung m.b.H., Karlsruhe

Abstract

In different parts of this report the computer programs developed in association with the Neutron Nuclear Data File KEDAK maintained in Karlsruhe are described. In this part the NDF-program package for the retrieval of the data from KEDAK and the program system KEMA for updating the KEDAK library e.g. for deletion, insertion, change of data records are presented. These programs are written in FORTRAN IV and are presupposing the direct access form of the KEDAK library as used in Karlsruhe.

Zusammenfassung

In den verschiedenen Teilen dieses Berichtes werden die Computerprogramme beschrieben, die in Karlsruhe in Verbindung mit dem dort unterhaltenen Neutronenkerndatenfile KEDAK entwickelt wurden. In diesem Teil werden die NDF-Subroutinen zum Lesen der auf KEDAK befindlichen Daten und das Programmsystem KEMA, das zur Aufdatierung der KEDAK Daten z.B. zum Löschen, Einfügen, Ändern von Datensätzen dient, dargestellt. Diese Programme sind in FORTRAN IV geschrieben und setzen die in Karlsruhe benutzte Direct Access Form der KEDAK Bibliothek voraus.

Contents

		Page
I.	INTRODUCTION	1
II.	DIRECT ACCESS KEDAK LIBRARY	2
III.	NDF - A RETRIEVAL PROGRAM FOR THE DIRECT ACCESS KEDAK LIBRARY	12
	1. The NDF retrieval mode	13
	2. The LDF retrieval mode	14
	3. The IDF retrieval mode	15
	4. Examples	16
IV.	PROGRAM SYSTEM KEMA	20
	1. Purpose of the program system KEMA	20
	2. The control program	23
	3. Conversion of the KEDAK library from direct access into sequential form	24
	4. Updating the sequential KEDAK library	26
	5. Conversion of the sequential KEDAK library into direct access form	31
	6. Programs for the management of test data	32
	7. Print out of selected sets of data of the sequential KEDAK library	33
	8. Print out of a list of contents of the direct access KEDAK library	34
	9. Conversion of the standard data of the KEDAK library from direct access form into card image format	35
	10. Conversion of the sets of data of selected isotopes from direct access form into card image format	36
	11. Conversion of the KEDAK library from card image format into direct access form	37
	12. Example for the input	38
V.	LITERATURE REFERENCES	40
VI.	APPENDIX: DATA TYPES FORESEEN ON KEDAK	41

I. Introduction

The data file KEDAK contains the evaluated neutron nuclear data for a number of materials important for reactor physics, specific physical experiments, burn up calculations, shielding and others. It is maintained and updated in Karlsruhe. It is used there in the direct access form whereas the sequential card image format is in use for the external exchange of the KEDAK library.

In this part the NDF-program package for the retrieval of the data stored on the KEDAK-file is described and also the program system KEMA for the management of the KEDAK library in particular for deletion, insertion, exchange of data records is presented here. Both are presupposing the direct access form of the KEDAK library. The nomenclature used in this description follows that described in KFK 880 [1]. For the direct understanding of the following the most important terms will be shortly explained.

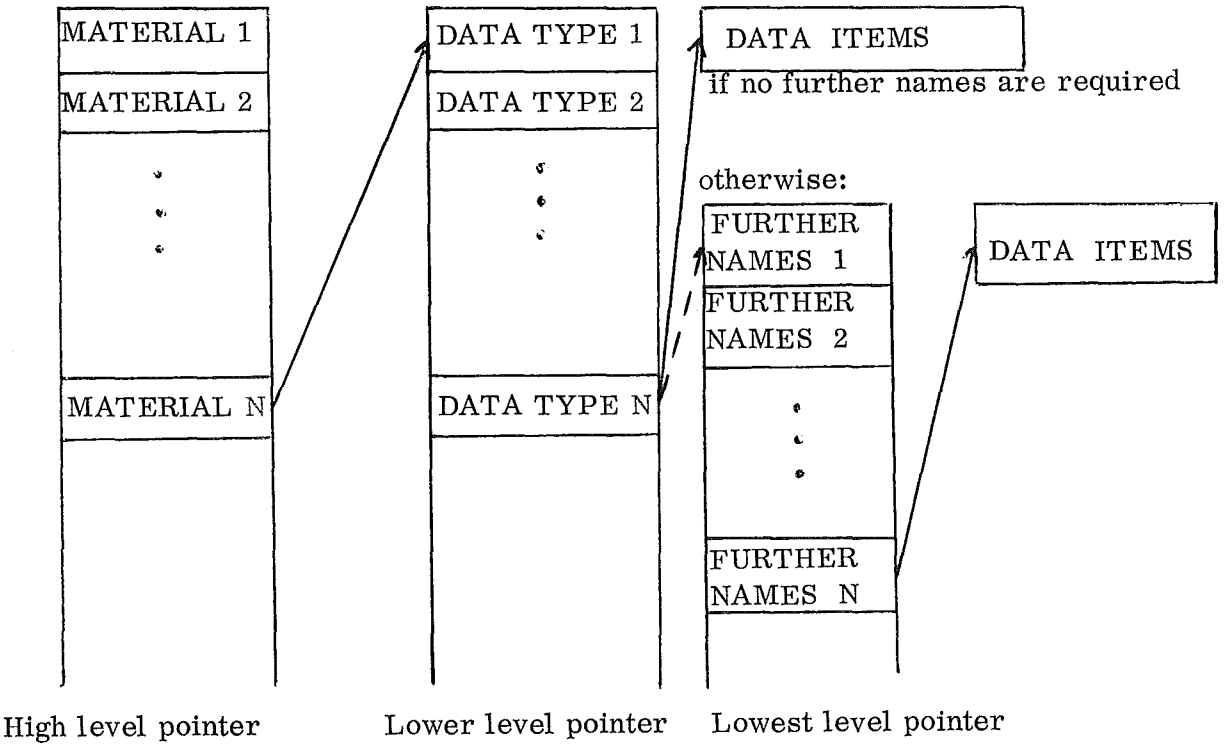
Each data type and each material is characterized by a numerical and an alphamerical name. Numeric names are used for storage purposes and in external transmission. The user of the direct access library alternatively may employ alphamerical names which might be kept in mind more easily since they are abbreviations containing the chemical symbols. For some data types additional so-called "further names" are necessary for their complete and unique description e.g. for the inelastic excitation cross section it is not sufficient to know its numerical data type name, but in addition the level excitation energy must be specified. The level energy is called a further name of this data type.
"Arguments" of the data type are called those quantities on which the data type considered is dependent and in dependence on which it is stored on the file, e.g. all cross section types have only one argument, the incident neutron energy. The numerical values of the nuclear data types corresponding to specific values of the arguments are termed "functional values". The number and kind of arguments and functional values for a single data item and also the number and kind of further names for a single data item are contained in a table of the Appendix for each of the data types at present available on KEDAK. This table supersedes the corresponding table in KFK 880 [1].

II. Direct access KEDAK library

The sets of data are written on a disk storage in direct access form without format control in form of fixed-length unblocked records. The entire information is divided into a declaration and a data part. All records are filled up to avoid unnecessary storage requirements. The structure of the declaration and data part is presented below in more detail.

Declaration part

The purpose of the declaration part is to provide necessary addressing information to access the data on the library. Addressing is performed by a hierarchy of pointers. Each pointer points to a table containing the next lower level of pointers. The lowest level pointer points to the starting address of a single data type.
An example may illustrate that: A table of pointers locates the starting address of the materials in the library. At this starting address another table provides the starting address of the various data types available for that material. If this data type requires further names, e.g. inelastic excitation levels, this address actually will point to a third table containing the starting address of the uniquely defined data types (see figure below).



Here the arrows indicate that the high level pointer contains the starting address of the next lower level pointer table or of data items.
The declaration part only includes the high level pointer table, the "material address table". As was pointed out before, numeric names are internally used to identify material or data type names. Therfore a link has to be established between the alphameric names and their numeric correspondent. This is obtained by two "conversion tables" stored in the declaration part.
In addition the declaration part contains an identifier for the library, its creation date and the information necessary to access the three tables described above.

Structure of the declaration part:

The declaration part starts at the first word of the first record in the library.

The content of the different records in the declaration part is outlined below.

Starting address[1] (record, word)	length (in words)	contents
1, 1	3	Library identification: the alphameric text 'KEDABIBLIOTH' is stored to enable identification of the library
1, 4	1	creation date: contains the date of the last update run for the library in decimal digits ddmmyy specifying day, month, year
1, 5	1	number of isotopes in the conversion table for material names ("material conversion table")
1, 6	2	starting address[1] of the "material conversion table"
1, 8	1	number of data type names appearing in the conversion table for data type names ("type conversion table")
1, 9	2	starting address[1] of the "type conversion table"
1, 11	1	number of materials in the table pointing to the starting address of each material ("material address table")
1, 12	2	starting address[1] of the "material address table"
as specified in the words 6 and 7	three times the number specified as contents in word 5	"material conversion table": consisting of three words for each material of which the first two contain alphameric material names of eight characters in length and the last one an integer number giving the numeric name: <table><tr><td>alphameric name of material 1</td><td></td><td>numeric name of material 1</td><td>alphameric name of material 2</td><td></td><td>numeric name of material 2</td><td>. . .</td></tr></table>
as specified in the words 9 and 10	three times the number given in word 8	"type conversion table": its structure is identical with that of the "material conversion table", so that in the above description only the term "material" has to be replaced by the term "data type"

as specified in the words 12 and 13	four times the number given in word 11	"material address table": this table points to the starting address of each material in the library. At this address the beginning of a table is found pointing to the individual data types for that material. Structure: the table consists of four words for each material. The first word gives the numeric material name, the second the number of data types available for that material, and the third and fourth word the starting address[1] for that material:

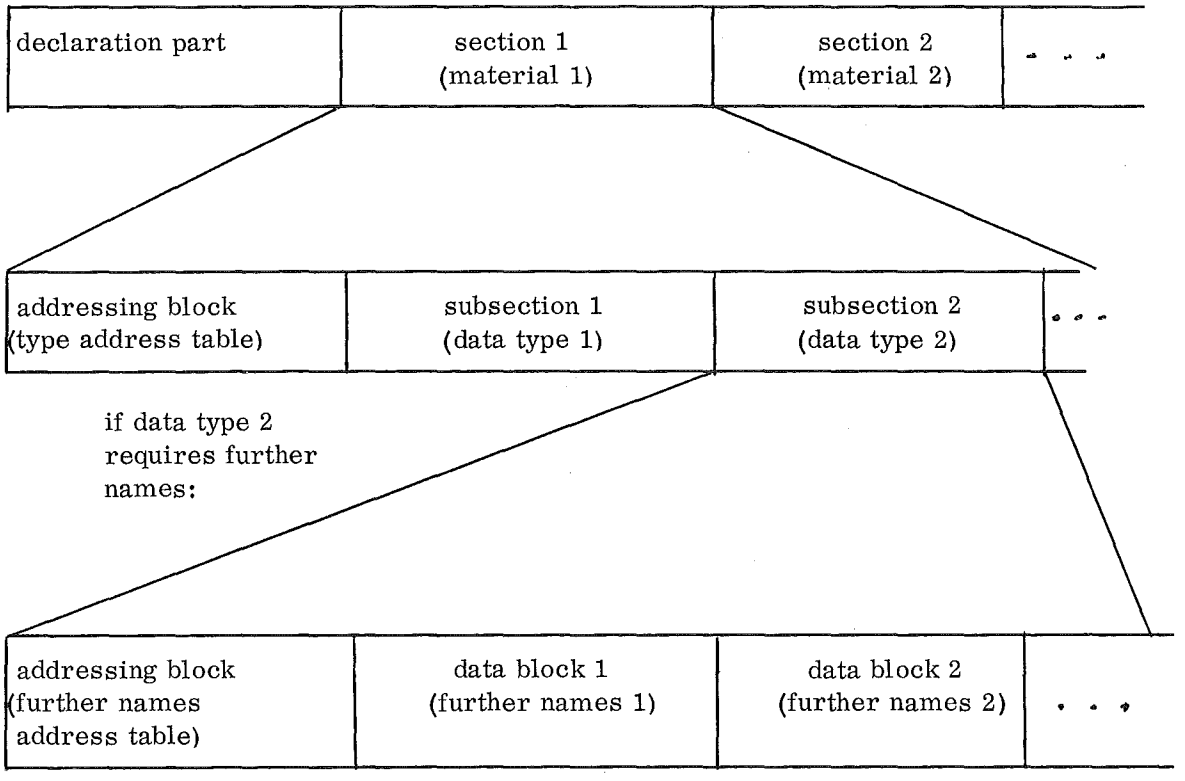
material 1				material 2				
numeric name	number of data types	starting address[1] of material 1 (record number)	(word number)	numeric name	number of data types	starting address[1] of material 2 (record number)	(word number)	

[1] An address in the library consists of two words: the first word gives the record number, the second the word number in this record.

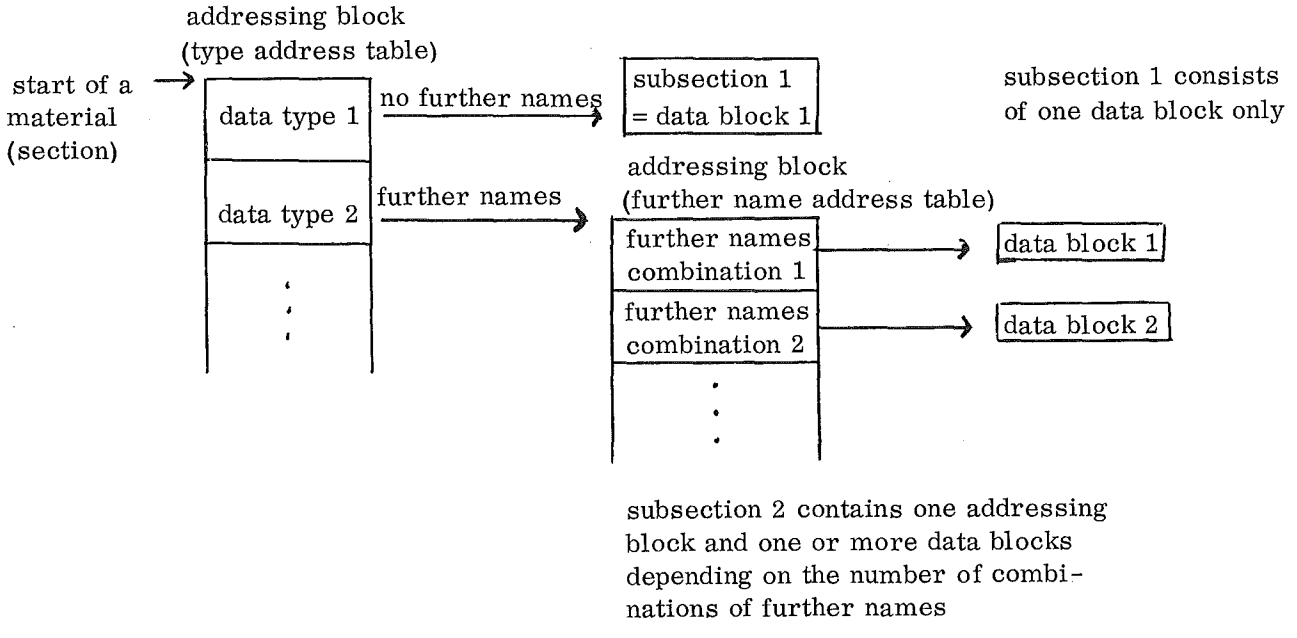
The contents of the declaration part is stored word by word consecutively in the above order. Since the starting address of the above three tables is explicitly specified in the words 5 - 13 of record one, this however is no prerequisite for the programs described in this report. Actually the tables could be stored anywhere within the library.

Data part

The data part includes all lower level pointer tables and the actual data items for the materials listed in the "material address table". The data part may be divided into sections, each section containing the information for a single material. As a rule a section is not physically interrupted by data of another section, that means each section is a closed block in the library.
Each section consists of an addressing block and subsections. The addressing block begins at the first word of the respective section. The corresponding starting address is stored in the material address table. A subsection contains the data for one data type and its starting address is recorded in the addressing block in a manner explained below. If the respective data type requires further names for unique identification an additional address block has to be given in the subsection. A subsection contains one or more data blocks depending upon whether further names do exist or not. The figure below may illustrate the physical organization:

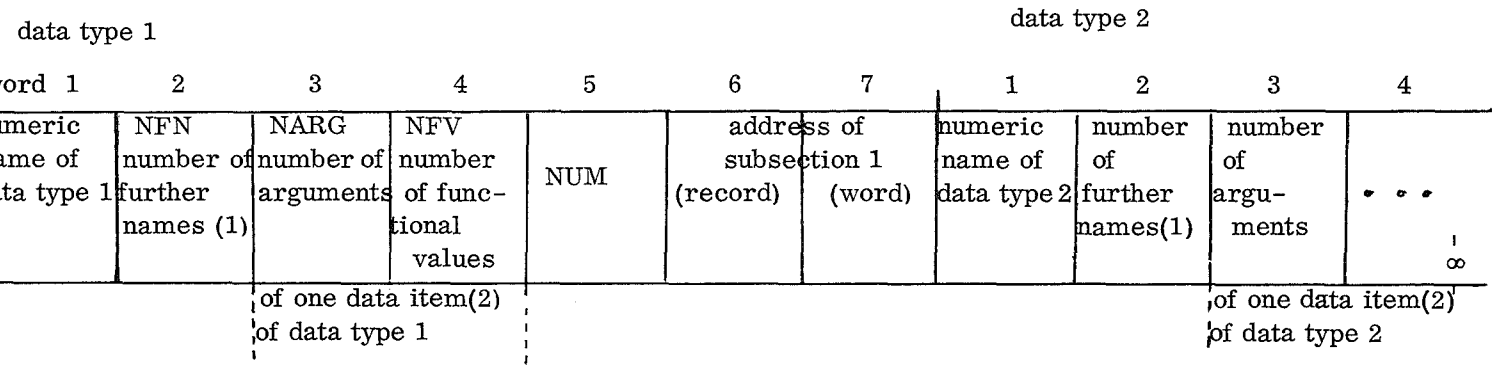


The logical structure of a section is displayed in the following figure:



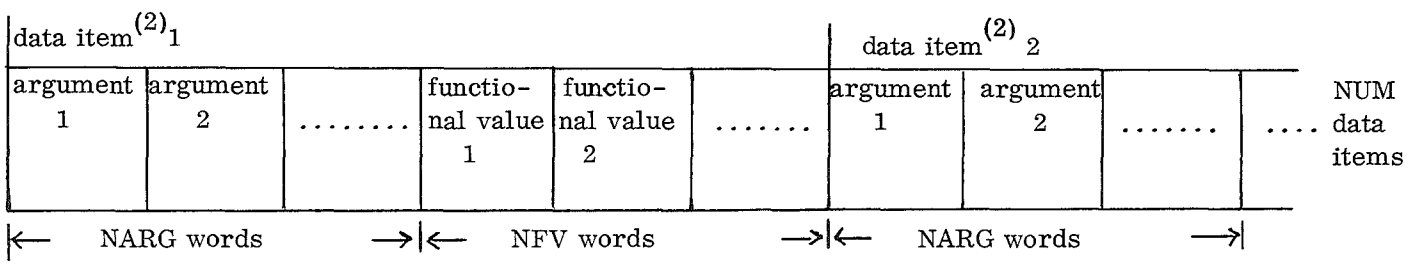
Structure of a section

starting address	length (in words)	contents
as given in the material address table	seven times the number of data types	"type address table": for each data type it points to the starting address of its subsection or the addressing block of this subsection if there is one. It consists of seven words for each data type with the following meaning:

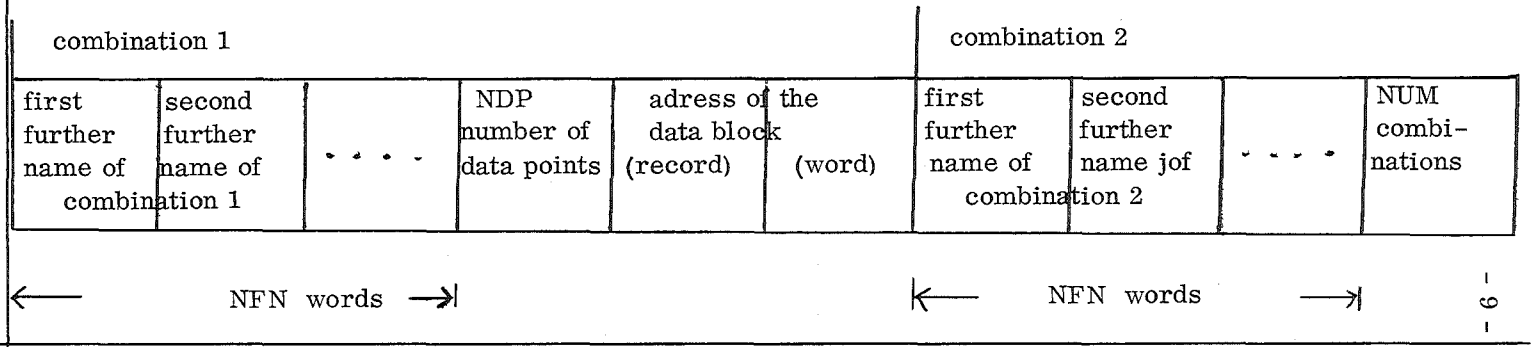


NUM: if no further names exist, NUM gives the number of data items for this data type
if further names exist, NUM gives the number of combinations of further names, e.g. for inelastic excitation the number of level energies for which data are given.

as given in the type address table	(NARG+NFV) * NUM where NARG, NFV and NUM are given in the type address table	subsection 1: assume, that data type 1 does not require further names, then its structure is that of a data block:



as given in the type address table	(3+NFN)*NUM where NFN and NUM are given in the type address table	Subsection 2: assume, that data type 2 does require further names, then the subsection starts with a "further name address table": for each combination of further names it contains the further names, the number of data points available and the starting address of the respective data block:
as given in the further name address table	(NARG+NFV)* NDP where NARG and NFV are given in the type address table and NDP is recorded in the further names address table	data blocks for the different combinations of further names follow. Their structure is similar to that of the data block described above



(1) The number of further names required for a data type are described in table 1 in the Appendix. Note the difference between number of further names and number of the combinations of further names: the number of further names required for inelastic excitation cross sections is one, the level energy, i.e. each combination of further names consists of one further name only. The number of the combinations of further names however, equals the number of level energies for which data are given.

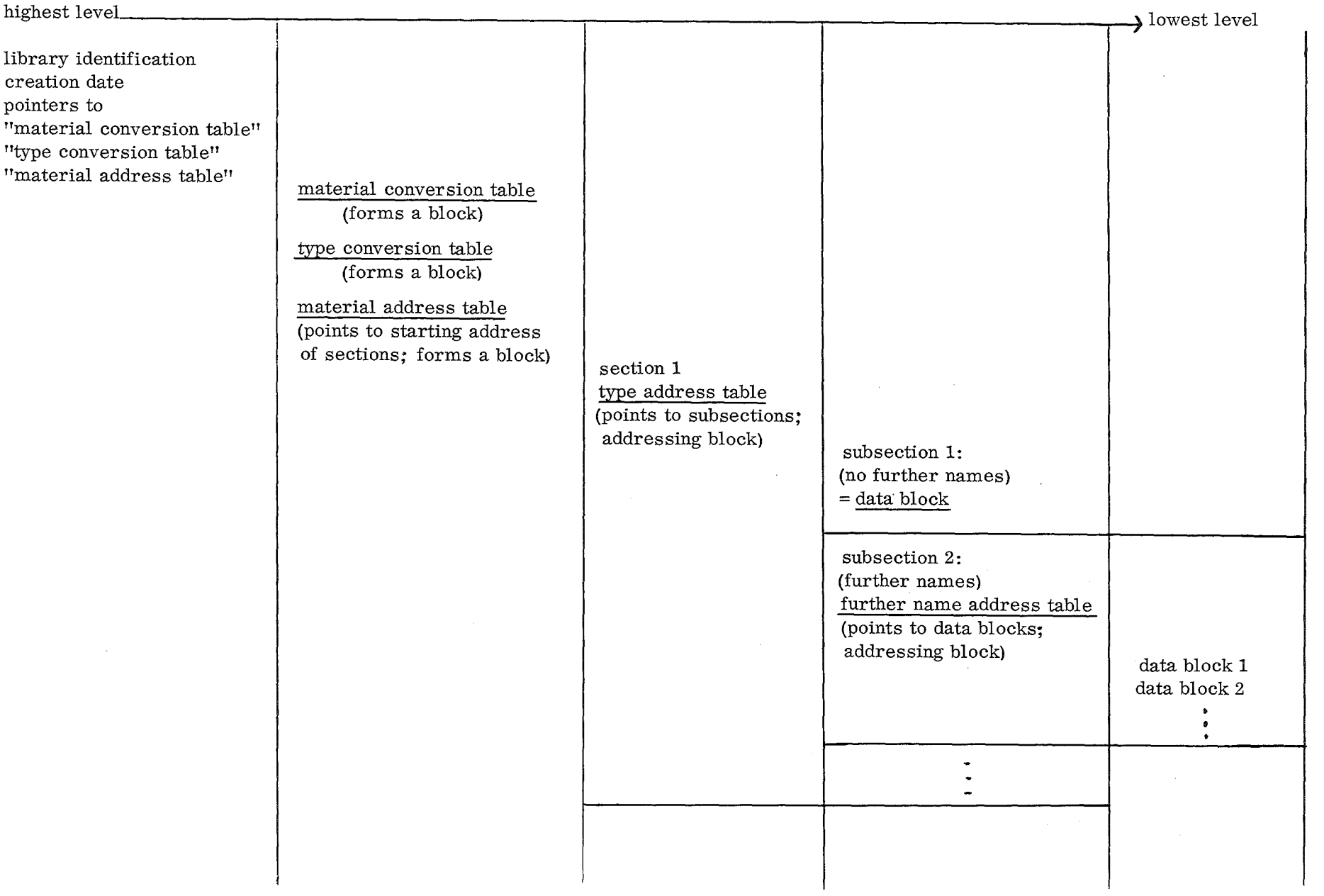
(2) The number of arguments and the number of functional values for each data type are given and interpreted in table 1 in the Appendix.
A data item is a set of arguments and functional values which logically belong together, e.g. for a cross section type the incident neutron energy (argument) and the cross section (functional value) form a data item. For details see [1].

Notes

Although data are stored densely in the library since the updating program described below does align them in this manner, this dense storage is not a prerequisite to use any of the programs described here. However a block may not be interrupted by data not belonging to it, no matter whether it is an addressing block or a data block. Any block may be shifted to any other free position in the library provided the next higher level pointer pointing to it is reset to the new address. The only exception are the words 1 - 13 of record 1, which represent the highest level block and must not be shifted.

Summary

The flow chart below gives an idea of the logical and physical organization of the library. The term "block" is used in the above sense.


highest level
lowest level
library identification
creation date
pointers to
"material conversion table"
"type conversion table"
"material address table"
material conversion table
(forms a block)
type conversion table
(forms a block)
material address table
(points to starting address
of sections; forms a block)
section 1
type address table
(points to subsections;
addressing block)
subsection 1:
(no further names)
= data block
subsection 2:
(further names)
further name address table
(points to data blocks;
addressing block)
data block 1
data block 2

III. NDF - a retrieval program for the direct access KEDAK library

The purpose of this FORTRAN VI retrieval routine is to allow Fortran programs to access the KEDAK library. The routine supplies three different modes for the retrieval of the data. Each of them consists of three entries into the subroutine NDF. Each retrieval mode performs the same basic task:

a) - OPN entries: the KEDAK library linked to the program by the DD-statement FTO1FOO1 is opened and its identifier, the first three words on record one of the library, is tested (see preceding section). If this test does not fail, the declaration part of the library is read into the main storage. The -OPN entry may be called only once in a program.

b) - LOC entries: locating information for the material and the daty type specified in the call is retrieved, and the type address table for that material is read into the main storage, if it is not already residing there. The first data item then is retrieved and passed to the user. If however the requested material or data type is not found in the library, a return code is set and no retrieval of data is attempted.
In case of further names the values specified by the user need not agree with those actually requested: a search is started on the further name address table and the first combination of further names is selected, for which each name is larger or equal to the corresponding one specified by the user. Let the user for example specify a level energy of zero. This will result in selection of the first level energy available for that material. The corresponding value (s) in the argument list (see next paragraph) is (are) changed to the selected one(s). In the above example the zero is changed to the value of the first level energy. If no combination of further names can satisfy the above requirement a return code is set and no retrieval of data is attempted.
Each time data for a new combination of names are required, a call to a -LOC entry must be executed.

c) - NXT entries: the data item immediately succeeding the one retrieved most recently is transmitted to the user. Note, that the first data item for a given combination of names must be accessed by the respective -LOC entry. If the most recently retrieved data item had been the last of the data type, a return code is set and no retrieval of data is attempted. If this condition is encountered, the information generally returned is undefined. In the case of further names however the next combination of further names is transmitted and may be used in a subsequent -LOC call.

The hyphen in the above notation must be replaced by NDF, LDF or IDF depending on the retrieval mode to be used. The retrieval modes and the argument lists of the respective entries are discussed in more detail below.

1. The NDF retrieval mode

This mode of the reading routine is retained only for historical reasons.

a) CALL NDFOPN (ARG 1, ARG 2, ARG 3, ARG 4)

ARG 1 Fixed point variable or constant giving the data set reference number ascribed to the KEDAK library. At present it has to be set equal to 1 and the DD-name used must be FTO1FOO1

ARG 2 Fixed point array of length two words. This array is filled by NDFOPN with 'bbbbbbbb'

ARG 3 Fixed point variable which is used by NDFOPN to pass the date of the last change of the KEDAK library in the form ddmmyy (see II.a)

ARG 4 Variable or constant without meaning

b) CALL NDFLOC (ARG 1, ARG 2, ARG 3, ARG 4, ARG 5)

ARG 1 Fixed point variable defining a return code to which the value O is assigned by NDFLOC if the requested data item was not found and otherwise the value 1

ARG 2 Fixed point array with a length of four words. The first word has to be specified in the calling program and gives the number of names of the required data set. The following three words are filled by NDFLOC with:

number of arguments for a single data item,
number of functional values for a single data item , O

ARG 3 Double precision array of adequate length. The first words have to to be specified in the calling program, in particular:
ARG 3(1) : name of the isotope in alphamerical form
ARG 3(2) : 'BEST bbbb'
ARG 3(3) : name of the data type in alphamerical form
In the case of further names the calling program must specify also:
ARG 3(4) : floating point value of the first further name
ARG 3(5) : floating point values of the other further names of a data set required
After searching the library for the first data item belonging to the specified names, NDFLOC stores the arguments and functional values of this first data item into ARG 3 starting with the first unused word.

ARG 4 Variable which is filled by NDFLOC with O
ARG 5 Variable which is filled by NDFLOC with O

c) CALL NDFNXT (ARG 1, ARG 2, ARG 3, ARG 4, ARG 5)

ARG 1 Return code. Is set 1 by NDFNXT if the last data item for the required data block has been retrieved, otherwise it is set equal to O

ARG 2 Fixed point array with a length of four words which are filled by NDFNXT with:
ARG 2 (1) : number of names of the set of data required
ARG 2 (2) : number of arguments of a single data item with these names
ARG 2 (3) : number of functional values of a single data item with these names
ARG 2 (4) : O

ARG 3 Double precision array of adequate length which is filled by NDFNXT successively with the names of the data set as specified in ARG 3 of NDFLOC, and the arguments and functional values of the data belonging to the above names and succeeding the one retrieved most recently

ARG 4 see NDFLOC

ARG 5 see NDFLOC

2. The LDF retrieval mode

This mode of the reading routine is the most refined one and its use is recommended therefore.

a) CALL LDFOPN (ARG 1, ARG 2, ARG 3)

ARG 1 Fixed point variable or constant giving the data set reference number ascribed to the KEDAK library. At present it has to be set equal to 1 and the corresponding DD-name is FTO1FOO1

ARG 2 Fixed point variable which is filled by LDFOPN with the date of the last change of the KEDAK library in the form ddmmyy (see II,a)

ARG 3 & Statement number
This statement number marks the position in the calling program at which the program execution should be resumed in case the names required in ARG 3 of LDFLOC could not be found on KEDAK

b) CALL LDFLOC (ARG 1, ARG 2, ARG 3, ARG 4)

ARG 1 Fixed point variable defining a flag which is filled by LDFLOC with O in case the required set of data was not found, otherwise filled with 1

ARG 2 Fixed point array with a length of three words. The first word has to be specified in the calling program and has to give the number of names of the required data set. The succeeding two words are filled by LDFLOC with:

number of arguments for a single data item,
number of functional values for a single data item

ARG 3 Double precision array of adequate length which the calling program has to fill with:
ARG 3 (1) : name of the isotope in alphamerical form
ARG 3 (2) : name of the data type in alphamerical form

and in the case of further names:

ARG 3 (3) : floating point value of the first further name
ARG 3 (4) : floating point values of the other further names
⋮ of the data set required

ARG 4 Floating point array of adequate length which is filled by LDFLOC with the first data item belonging to the names supplied in ARG 3

c) CALL LDFNXT (ARG 1, ARG 2, ARG 3, ARG 4)

ARG 1 Return code. Is set 1 by LDFNXT if the last data item for the required data block has been retrieved, otherwise it is set equal to O

ARG 2 Fixed point array with a length of three words which are filled by LDFNXT with:
ARG 2 (1) : number of names of the set of the data required
ARG 2 (2) : number of arguments for a single data item with these names
ARG 2 (3) : number of functional values for a single data item with these names

ARG 3 Double precision array of adequate length which is filled by LDFNXT successively with the names of the data set in alphamerical form as specified in ARG 3 of LDFLOC

ARG 4 Floating point array of adequate length which is filled by LDFNXT with the data item next to the one retrieved most recently.

3. The IDF retrieval mode

This mode of the reading routine has compared with the LDF mode the preference of a shorter searching time in the library, since the isotope and data type names have to be given in the internal numerical form. The reading of data sets is performed in the following way:

a) CALL LDFOPN (ARG 1, ARG 2, ARG 3)
see III.a

b) CALL IDFLOC (ARG 1, ARG 2, ARG 3, ARG 4)

ARG 1 see LDFLOC III.2 b

ARG 2 see LDFLOC III.2 b

ARG 3 single precision array of adequate length which is filled by the calling program with:
ARG 3 (1) : name of the isotope in numerical form
ARG 3 (2) : name of the data type in numerical form

In the case of further names:
ARG 3 (3) : floating point value of the first further name
ARG 4 (4) : floating point values of the other further names
of the set of the data required

ARG 4 see LDFLOC III.2 b

c) CALL IDFNXT (ARG 1, ARG 2, ARG 3, ARG 4)

ARG 1 see LDFNXT III.2 c

ARG 2 see LDFNXT III.2 c

ARG 3 single precision array of adequate length which is filled by IDFNXT with the names of the set of data in numerical form

ARG 4 see LDFNXT III.2 c

4. Examples

a) Reading and printing of the data between EU and EO for the fission cross section (KEDAK type SGF) of PU 239
EU = 30 KeV
EO = 70 KeV

```
      REAL*8 MAT,TYP,NAM
      DIMENSION E(300),SIGMA(300),NNAM(3),NAM(2),QUER(2)
      DATA MAT/'PU239   '/,TYP/'SGF     '/
      EU=30000.
      EO=70000.
      I=1
      CALL LDFOPN(1,NDATUM,&5)
      WRITE(6,1) NDATUM
    1 FORMAT(' DATE OF THE KEDAK LIBRARY',I10//)
      NNAM(1)=2
      NAM(1)=MAT
      NAM(2)=TYP
      CALL LDFLOC(KENNZ,NNAM,NAM,QUER)
      IF(KENNZ.EQ.0)  GO TO 2
      IF(QUER(1).LT.EU)  GO TO 3
   12 E(I)=QUER(1)
      SIGMA(I)=QUER(2)
      I=I+1
      IF(I.LE.300)  GO TO 3
      GO TO 5
    2 WRITE(6,4) MAT,TYP
    4 FORMAT(' FOR THE ISOTOPE ',A6,' THE DATA TYPE ',A7,' IS NOT AVAILA
     1BLE IN THE KEDAK LIBRARY')
      GO TO 11
    3 CALL LDFNXT(KENNZ,NNAM,NAM,QUER)
      IF(KENNZ.EQ.0)  GO TO 5
      IF(QUER(1).LT.EU)  GO TO 3
      IF(QUER(1).GT.EO)  GO TO 5
      GO TO 12
    5 I=I-1
      IF(I.GT.1)  GO TO 6
      WRITE(6,7) MAT,TYP,EU,EO
    7 FORMAT(' FOR THE ISOTOPE ',A6,' AND THE DATA TYPE ',A6,' NO DATA I
     1TEMS ARE AVAILABLE IN THE ENERGY RANGE FROM',E16.8,'EV TO',
     2E16.8,'EV')
      GO TO 11
    6 WRITE(6,8)
    8 FORMAT(6X,'ENERGY',11X,'SGF')
      WRITE(6,9)(E(J),SIGMA(J),J=1,I)
    9 FORMAT(2E16.8)
   11 STOP
      END
```

Output :

```
 DATE OF THE KEDAK LIBRARY    100571


      ENERGY            SGF
  0.30000000E 05  0.16974993E 01
  0.35000000E 05  0.16780996E 01
  0.40000000E 05  0.16570997E 01
  0.45000000E 05  0.16438999E 01
  0.50000000E 05  0.16248999E 01
  0.55000000E 05  0.16062994E 01
  0.57000000E 05  0.16004000E 01
  0.60000000E 05  0.15901995E 01
  0.65000000E 05  0.15709991E 01
  0.70000000E 05  0.15663996E 01
```

b) Reading and printing of the data for the inelastic excitation cross section (KEDAK type SGIZ) of the excited levels between E 1 and E 2 for U 238

E 1 = O

E 2 = 200 KeV

```
      REAL*8 MAT,TYP,NIVEAU,NAM
      DIMENSION E(300),SIGMA(300),NNAM(3),NAM(3),QUER(2)
      DATA MAT/'U 238   '/,TYP/'SGIZ    '/
      E1=0.
      E2=200000.
      I=1
      CALL LDFOPN(1,NDATUM,&3)
      WRITE(6,1)NDATUM
    1 FORMAT(' DATE OF THE KEDAK LIBRARY',I10//)
      NNAM(1)=3
      NAM(1)=MAT
      NAM(2)=TYP
      NAM(3)=0.
      CALL LDFLOC(KENNZ,NNAM,NAM,QUER)
      IF(KENNZ.EQ.1)  GO TO 2
    5 CALL LDFLOC(KENNZ,NNAM,NAM,QUER)
      IF(KENNZ.EQ.0)  GO TO 3
    2 IF(NAM(3).GE.E1)  GO TO 9
      NAM(3)=NAM(3)*1.0001
      GO TO 5
    3 WRITE(6,6) MAT,TYP
    6 FORMAT(' FOR THE ISOTOPE ',A6,' THE DATA TYPE ',A7,' IS NOT AVAILA
     1BLE IN THE KEDAK LIBRARY')
      GO TO 11
    9 NIVEAU=NAM(3)
      IF(NAM(3).GT.E2)  GO TO 11
    4 E(I)=QUER(1)
      SIGMA(I)=QUER(2)
      I=I+1
      IF(I.GT.300)  GO TO 14
      CALL LDFNXT(KENNZ,NNAM,NAM,QUER)
      IF(KENNZ.EQ.1)  GO TO 4
   14 I=I-1
   10 WRITE(6,7) NIVEAU
    7 FORMAT(//'  LEVEL ENERGY :',E16.8//6X,'ENERGY',11X,'SGIZ')
      WRITE(6,8) (E(J),SIGMA(J),J=1,I)
    8 FORMAT(2E16.8)
      I=1
      IF(NAM(3).GT.NIVEAU.AND.NAM(3).LE.E2)  GO TO 5
   11 STOP
      END
```

Output:

```
DATE OF THE KEDAK LIBRARY     100571


 LEVEL ENERGY :   0.44700000D 05

     ENERGY              SGIZ
 0.39000000E  05   0.0
 0.44900000E  05   0.0
 0.47000000E  05   0.99999979E-02
 0.50000000E  05   0.49999997E-01
 0.53100000E  05   0.79999983E-01
       .
       .
       .

 0.14500000E  07   0.12699997E 00
 0.14700000E  07   0.93399942E-01
 0.15000000E  07   0.42999998E-01
 0.15500000E  07   0.0
 0.16000000E  07   0.0
 0.16500000E  07   0.0
 0.17000000E  07   0.0
 0.17500000E  07   0.0
 0.18000000E  07   0.0
 0.18500000E  07   0.0
 0.19000000E  07   0.0
 0.19500000E  07   0.0
 0.20000000E  07   0.0


 LEVEL ENERGY :   0.14800000D 06

     ENERGY              SGIZ
 0.14000000E  06   0.0
 0.14800000E  06   0.0
 0.16000000E  06   0.12999997E-01
 0.18000000E  06   0.36999997E-01
 0.20000000E  06   0.57999998E-01
       .
       .
       .

 0.17000000E  07   0.97999990E-01
 0.17500000E  07   0.61999999E-01
 0.18000000E  07   0.22000000E-01
 0.18500000E  07   0.0
 0.19000000E  07   0.0
 0.19500000E  07   0.0
 0.20000000E  07   0.0
```

IV. Program System KEMA

1. Purpose of the program system KEMA

The purpose of KEMA is to execute all kinds of management tasks on the KEDAK library. These can be all possible changes of nuclear data which imply a previous transformation from direct access form to a sequential form; the generation of the KEDAK library in card image format for dispatching the data to other centers; the retransformation from card image format to direct access form and the printout of KEDAK summary information.
KEMA also provides a number of routines for the management of the test data sets in the library. Similarly to the existence of ENDF/A and ENDF/B KEDAK offers the possibility to enter incomplete or not fully evaluated sets of data, e.g. to test new microscopic experimental results. Since these data are not distributed to external users, the organization of these sets of data and their management will not be described in this publication, although it forms part of the KEMA program system. Each of the above tasks will be executed by respectively one subroutine which is called by a control program. Therefore this FORTRAN VI program system is always expansive. The separate subroutines will be loaded in the program region by an overlay structure only if they are going to be used. The control program extracts from the input what kind of task will be done, checks the input of all desired working programs and the presentation of all needed DD-cards and calls the appropriate subroutines.
The entire input for the called subroutines is interpreted and written on unit 8 in unformatted form by a FORTRAN VI subroutine FREEFO. The working programs of KEMA can then read their particular input from unit 8. For the input some explanations are necessary: Each data record starts in column 1 of a data card. If it is not possible to place all the data of one input record on one card, a second, third, etc. card may be used, which must have a blank in column 1. Or: a non-blank column 1 in the input card is an indication for a new input record.

One has to distinguish between an alphameric word of the length REAL*4 and of the length REAL*8. A REAL*4 word must be included in apostrophies and is stored left justified in the computer and filled up with blanks if not all 4 bytes are occupied. Example: 'ARG' or 'ABCD'. A REAL*8 word may be included in apostrophies comprising at least 5 signs and at most 8 signs, which are stored left justified in the computer if not all 8 bytes are occupied. A special case are REAL*8 words with a number of occupied bytes less than or equal to 5. These words may be also included in @-signs. They are also stored left justified in the computer and filled up with blanks.
Examples: 'PU 239' ≡ 'PU 239bbb' ≡ @PU 239@ ; 'RESbb' ≡ @RES@ .
Fixed point and floating point numbers are written in the usual manner, e.g: fixed point numbers: 1 10 875 and floating point numbers: 10. 5.E3 0.7E-3 0.01. It is not possible to read double precision values.

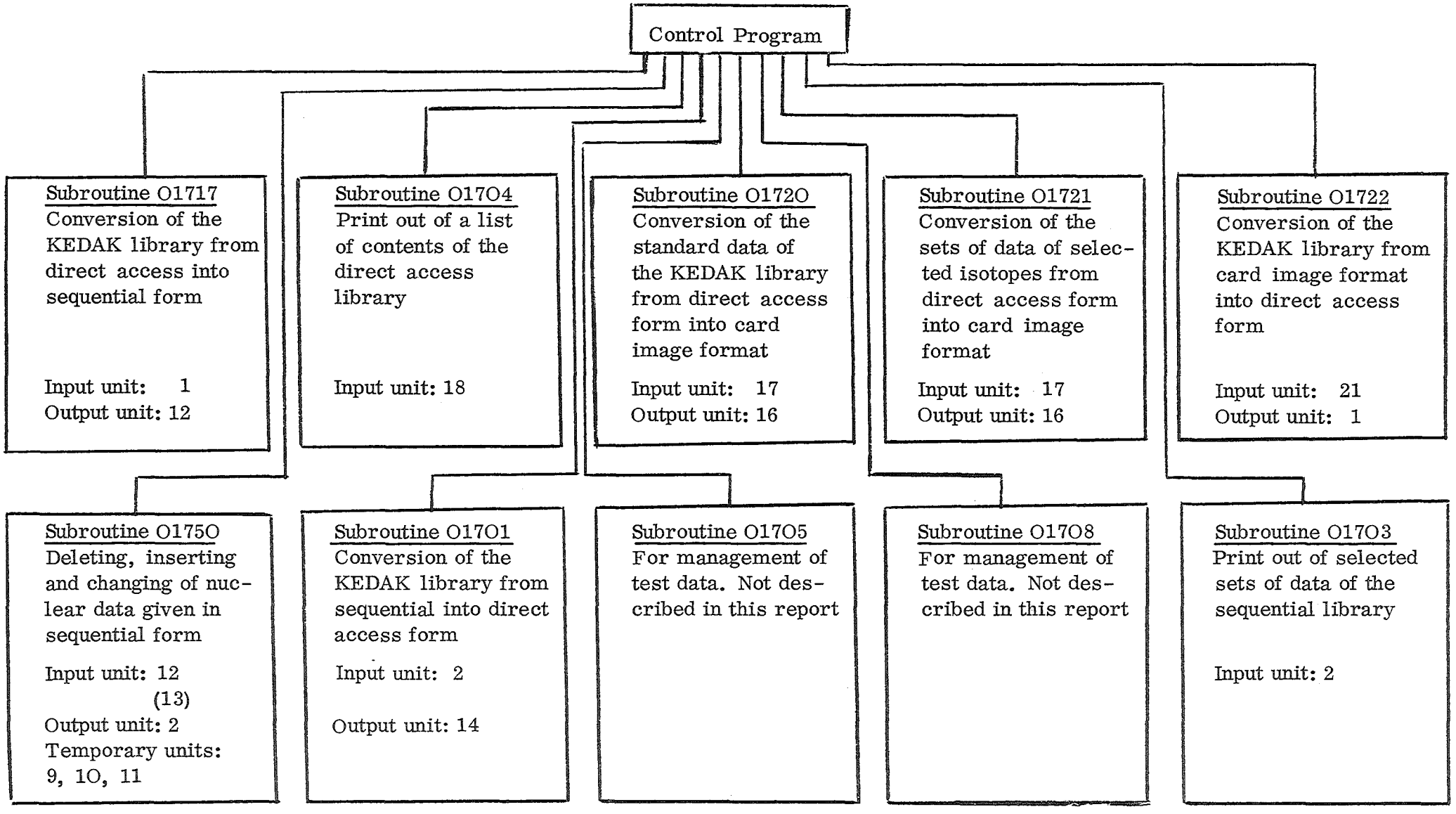
The input data are separated by one or more blanks.

The following subroutines are contained in KEMA:

Identification number		Task of the respective program
O1717	-	Program for converting the KEDAK library from direct access into sequential form
O175O	-	Program for deleting, inserting and changing nuclear data which are given in sequential form
O17O1	-	Program for converting the KEDAK library from sequential into direct access form
O17O5	-	Program for filling up data gaps in test data sets by standard data in the sequential KEDAK library
O17O8	-	Program for declaration of test data to standard data in the sequential KEDAK library
O17O3	-	Program for printing selected sets of data of the sequential KEDAK library
O17O4	-	Program for printing a list of contents of the direct access KEDAK library
O172O	-	Program for converting standard data of the KEDAK library from direct access form into card image format
O1721	-	Program for converting the sets of data of selected isotopes of the KEDAK library from direct access form into card image format
O1722	-	Program for conversion of the KEDAK library from card image format into direct access form

For updating the direct access KEDAK library by standard data, only the programs O1717, O175O and O17O1 are needed.

Diagram of the program system KEMA



2. The control program

Input:

1. record (containing the information about the program flow)

   I Number of working subroutines to be called + 1 ($I \leq 20$)

   (NFOLG(J), J = 1, I) Identification numbers of the working subroutines chosen. NFOLG (I) = O

2. record (containing the assignment table of the alphanumerical and the numerical names of the isotopes)

   NZM Number of isotopes in the assignment table

   (MATNA(J), NUNA(J), J = 1, NZM)

   MATNA : alphanumerical name of the isotope (REAL* 8)
   NUNA : numerical name of the isotope

3. record (containing the assignment table of the alphanumerical and the numerical data type names)

   NZT Number of data types in the assignment table

   (TYPN(J), NUTY(J), J = 1, NZT)

   TYPN : alphanumerical data type name (REAL* 8)
   NUTY : numerical data type name

3. Conversion of the KEDAK library from direct access into sequential form

Program : O1717

Input : KEDAK library in direct access form (organization see II) on the external storage unit 1

Output : KEDAK library in sequential form on the external storage unit 12

This sequential library has the following unformatted organization:

1. record

1 constant,
date of the last change in the form ddmmyy,
number of isotopes available in the library.

2. record

Names of the isotopes in numerical form
The following records are repeated for each isotope

3. record

Name of the isotope in numerical form,
number of data types available for this isotope

4. record

Data type names in numerical form
The following records are repeated for each data type name

5. record

Name of the isotope in numerical form,
name of the data type in numerical form,
number of further names of the data type,
number of arguments for a single data item,
number of functional values for a single data item
in the case of further names:
number of combinations of the further names
otherwise:
O

The following records are repeated for each combination of further names. If there are no further names at all the 6^{th} record is omitted and the other two records are given only once.

6. record

Further names of the combination.

7. record

Number of data items.

8. record

Arguments for a single data item,
functional values belonging to the arguments for a single data item for all data items in the order of increasing arguments.

4. Updating the sequential KEDAK library

Program : 175O

Input : a) Sequential KEDAK library (organization see IV 3) on the external storage unit 12

b) Control input in form of card input

c) Data blocks for changing existing data either in the form of card input or in the form of a data set on the external storage unit 13

The control input has the following structure:

1. record

I AU { In the case the data alteration blocks are given in form of cards: 1
otherwise: O

I BA { In the case the data alteration blocks are given in form of a data set on the external storage unit 13: 1
otherwise: O

I AU and I BA may both be set equal to 1. In this case first the card input and then the external input is processed.

NDTUM Current date. This date appears in the first record of the new version of the sequential library.

NNKO Number of combinations of material / data type names, for which new combinations of further names shall be inserted.

2. record (only if NNKO> O)

((IKO (I, J), I = 1,2), IZKO (J), J = 1, NNKO)

IKO (1, . . .) : name of the isotope in alphanumerical form,
IKO (2, . . .) : name of the data type in alphanumerical form,
IZKO : number of combinations of new further names to be inserted.

Then the data alteration blocks follow. According to the kind of data change one has to choose between three types of data alteration blocks: ADD, DROPS and DROPA. The ADD-set has the function to insert single data items for the specified material and data type at the point specified by the respective arguments, the DROPS-set to delete single data items existing for the specified material and data type and identified by their arguments. The DROPA-set causes dropping for all data items for the specified material or only all data items for a specific data type. In arranging the data within the data alteration blocks first priority is assigned to the names of the isotopes. At present the following order of material names is valid for KEDAK :

NI_{bbb}, NI_b58, NI_b6O, NI_b61, NI_b62, NI_b64, $\emptyset_{bb}16$,
U_b235, AL_b27, $C_{bb}12$, CD_{bbb}, CR_{bbb}, CR_b5O, CR_b52,
CR_b53, CR_b54, FE_{bbb}, FE_b54, FE_b56, FE_b57, FE_b58,
$H_{bbb}2$, $H_{bb}H\,1$, $H_{bb}\emptyset\,1$, $HE_{bb}3$, $HE_{bb}4$, $M\emptyset_{bbb}$, $M\emptyset_b92$,
$M\emptyset_b94$, $M\emptyset_b95$, $M\emptyset_b96$, $M\emptyset_b97$, $M\emptyset_b98$, $M\emptyset\,1OO$, N_{bbbb},
NA_b23, PU239, PU24O, PU241, PU242, U_b238, $H_{bbb}1$,
$CL_{bbb}UNC$, CL_b35, CL_b37,

The actual order of material names can be printed out by the program O17O4 in IV. 8.
Second priority have the data type names which have to be arranged in alphabetical order. If there are further names for the data type considered the different combinations of further names are given in the order of increasing values. For one particular material, data type and one particular combination of further names (if required) the rewised data have to be given in order of increasing arguments.

Structure of the data alteration blocks

A Insertion of new data (ADD - blocks)

This item covers the three following tasks:

a) The names specified (material - and / or data type - and / or further names) are not found in the KEDAK library - a new set of data characterized by these names has to be created.

b) The names specified are already available in the library - new data items have to be inserted.

c) The specified names and arguments of single data items exist already in the library - the existing functional values have to be replaced by new values.

Structure of the data block:

N	Number of data words in the following input record (N≤ 2OOO). Note that alphameric names consist of eight characters each and have to be counted as two words.
∂ ADD_{bb} ∂	Constant, REAL* 8 word
NNAM	Number of names
(NAM(I), I = 1, NNAM)	Name of the isotope in alphanumerical form (REAL* 8), name of the data type in alphanumerical form (REAL* 8), eventual further names in single precision floating point representation.

NARG	Number of arguments of a single data item	
NWERT	Number of functional values of a single data item	
(ARG 1 (I), I = 1, NARG)	Arguments	first
(WERT 1 (I), I = 1, NWERT)	Funtional values	data item
(ARG 2 (I), I = 1, NARG)	Arguments	second
(WERT 2 (I), I = 1, NWERT)	Functional values	data item

and so on in the order of increasing arguments as many times as data items shall be inserted or changed. In the case the data type for which changes should be performed has no arguments at all (as it is the case e.g. for the type ISOT1, for other examples see [1] then NARG = 0 and there can only be one data item with ARG (1) equal to the alphameric text 'ARG' and functional values following.

The data type AASTATUS, too, represents a special case since it is only used to store bibliographic information. If this type shall be inserted, NARG and NWERT have to be set equal to 1 and the comments consequently following have to be included in apostrophies. They have to be given in total by a number of characters divisible by 8, since the first 4 characters are always interpreted as argument and the following 4 characters as functional value. Note, that 72 characters are thought to form one line in actual use.

B Deletion of existing data (DROPA -, DROPS - blocks)

a) The deletion of an entire set or sets of data on KEDAK is effected by the DROPA date block.
Structure of the data block:

N	Number of data words in the input record (N≤ 2000). (REAL*8 data have to be counted as two single words)
'DROPA'	Constant, (REAL*8)
NNAM	Number of names
(NAM (I), I = 1, NNAM)	Name of the isotope in alphanumerical form (REAL*8), eventually: name of the data type to be deleted in alphanumerical form (REAL*8), eventually: further names in floating point representation.

NNAM, (NAM (I), I = 1, NNAM) may be repeated as often as necessary.

b) The deletion of single data items belonging to the specified names and arguments is caused by the DROPS-block.

Structure of the DROPS-block:

N	Number of data words in the input record ($N \leq 2000$). (REAL* 8 words have to be counted as two single words)
∂ DROPS ∂	Constant, REAL* 8 word
NNAM	Number of names
(NAM(I), I = 1, NNAM)	Name of the isotope in alphanumerical form (REAL* 8), name of the data type in alphanumerical form (REAL* 8), eventually: further names in floating point representation
NARG	Number of arguments for a single data item
(ARG 1(I), I = 1, NARG)	Arguments of the first data item to be deleted
(ARG 2(I), I = 1, NARG)	Arguments of the second data item to be deleted

and so on in the order of increasing arguments as many times as data items shall be deleted.

The last input record of program O175O is given by:

2 ∂ $ENDE_b$ ∂

In the case that IAU = 1 and IBA = 1, i.e. data alteration blocks are presented in form of cards as well as on an external storage unit, first the data alteration blocks on cards closed by the 2 ∂ $ENDE_b$ ∂ record are processed and successively the external input which has also to be completed by the record 2 ∂ $ENDE_b$ ∂ . The hierarchy of the input data, however, i.e. the already specified order of the material names, the alphabetic order of the data type names and the increasing order of arguments has not only to be fulfilled within the data alteration blocks on cards and on an external unit but it has also to be valid for the entire input in the case both input variants are mixed.
The different data alteration blocks ADD, DROPA, DROPS are sorted by the program onto the different storage units 9, 1O and 11 respectively. Therefore DD-cards for these units are needed.

Output:

a) Sequential KEDAK library containing the changed data on the external storage unit 2 (organization see IV. 3)

b) A listing for each isotope and each data type, giving the
number of data items available,
number of data items deleted,
number of data items inserted,
number of data items exchanged.

5. Conversion of the sequential KEDAK library into direct access form

Program : O17O1

Input :

a) Sequential KEDAK library on the external storage unit 2 (organization see IV. 3)

b) The card input consists of one record with the following contents:

'KEDAK' 'BIBL' 'IØTH' Constants which appear as words 1, 2, 3 in the declaration part of the direct access KEDAK library (see II)

IDAT — Current date in the form ddmmyy. This date appears as the 4th word in the declaration part of the direct access KEDAK library

NMAT — Number of isotopes available in the sequential library

Output :

a) KEDAK library in direct access form (see II) on the external storage unit 14

b) Printed test output, e.g.

KMAT — Current number of the isotope which is just going to be converted

ITYP — Current number of the data type of this isotope KMAT which is just going to be converted

INK — Current number of the combination of further names belonging to ITYP and KMAT which is just going to be converted.

6. Programs for the management of test data

Programs : O17O5
O17O8

These programs allow to operate upon test data in the library (see IV. 1) and will not be described here.

7. Print out of selected sets of data of the sequential KEDAK library

Program : O17O3

Input : a) Sequential KEDAK library on the external storage unit 2 (organization see IV. 3)

b) Card input with the following contents:
N, (MAT (I), TYP (I), I = 1, N)

N : Number of isotope name/data type name combinations to be printed

MAT : Name of the isotope in alphanumerical form (REAL* 8)

TYP : Name of the data type in alphanumerical form (REAL* 8)

Here the order of isotope names as specified in IV. 4 and the alphabetic order of the data type names has to be observed.

Output : For each selected combination of names:

Name of the isotope,
name of the data type
(the following is repeated for each combination of further names, if existing for the respective data type)
further names if existing,
number of data items,
arguments, } for the first data item and so on
functional values } for all data items successively

8. Print out of a list of contents of the direct access KEDAK library

Program : O17O4

Input : KEDAK library in direct access form (organization see II) on the external storage unit 18

Output : A list of the contents of the KEDAK library, i.e. a list of the isotopes in the order as stored in the library and for each isotope the available alphanumerical data type names in alphabetical order.

9. Conversion of the standard data of the KEDAK library from direct access form into card image format

Program : O172O

Input : KEDAK library in direct access form (organization see II) on the external storage unit 17

Output : The sets of data of all the isotopes in the KEDAK library having names of a length less than or equal 5 characters in card image format are written onto the external storage unit 16. The organization of the KEDAK library in card image format is described in [1], except that the structure of the identification part was modified. The identification part contains now in columns

73 - 74 the position at which the isotope appears in the description of the isotope contents
or
O, if the record belongs to the description of the isotope contents

75 - 76 the position at which the data type appears in the isotope dependent description of the data type contents

or
O, if the record belongs to the description of the data type contents of a specific isotope name

77 - 8O the record count for a set of data of a specific isotope name, a specific data type name and eventually a particular combination of further names starting with O

Refer to [1] for an explanation of the terms used here.
An exception to the format codes described in [1] and used to prepare the card image library is the comment type AASTATUS, which is written in format (18A4)

10. Conversion of the sets of data of selected isotopes from direct access form into card image format

Program : O1721

Input : a) KEDAK library in direct access form (organization see II) on the external storage unit 17

b) Card input with the following contents:
(NNM, (TMAT (I), I = 1, NNM))

NNM : Number of isotopes the sets of data of which should be converted into card image format

TMAT : Names of these isotopes in alphanumerical form in the order given in IV. 4

Output : The sets of data of the isotopes specified in the input are written in card image form on the external storage unit 16. The output of program O1721 corresponds to the output of program O172O described in IV. 9.

11. Conversion of the KEDAK library from card image format into direct access form

Program : O1722

Input : a) The KEDAK library in card image format (organization see II) on the external storage unit 21. If the card image library is distributed on more than one magnetic tape, for each of these tapes a separate DD-card is necessary on which the unit numbers are continuously incremented by one starting with 21. If for example the nuclear data are distributed on 3 magnetic tapes:

```
//G.FT21FOO1  DD  UNIT=TAPE9, VOL=SER=TAPE1,...
//G.FT22FOO1  DD  UNIT=AFF=FT21FOO1, VOL=SER=TAPE2,...
//G.FT23FOO1  DD  UNIT=AFF=FT21FOO1, VOL=SER=TAPE3,...
```

b) The card input consists of one record with the following contents:

'KEDA' 'BIBL' 'ØTH'	Constants which appear as words 1, 2, 3 in the declaration part of the direct access KEDAK library (see II)
IDAT	Current date in the form ddmmyy. This date appears as the 4^{th} word in the declaration part of the direct access KEDAK library
NMAT	Number of isotopes available in the card image library
IBND	Number of magnetic tapes on which the card image library is described

Output : KEDAK library in direct access form as discussed in II on the external storage unit 1

12. Example for the input

The direct access KEDAK library shall be modified and the new library shall be converted into card image format

```
//INRO17KE_b JOB_b (0017, 101, P6M1A), KRIEG, CLASS=A, REGION=270 K,
//_bTIME=15
//_bEXEC FHG, LIB=NUSYS, NAME=KEMA
//G.FT08F001   DD   UNIT=SYSDA, SPACE=(TRK,10)
//G.FT01F001   DD   UNIT=2314, VOL=SER=NUSYS0,DSN=KNDF, DISP=SHR
//G.FT12F001   DD   UNIT=SYSDA, SPACE=(TRK,300,RLSE),
//_bDCB=(RECFM=VBS, BLKSIZE=7168)
//G.FT09F001   DD   UNIT=SYSDA, SPACE=(TRK,20)
//G.FT10F001   DD   UNIT=SYSDA, SPACE=(TRK,20)
//G.FT11F001   DD   UNIT=SYSDA, SPACE=(TRK,20)
//G.FT02F001   DD   UNIT=SYSDA, SPACE=(TRK,300,RLSE),
//_bDCB=(RECFM=VBS, BLKSIZE=7168)
//G.FT14F001   DD   UNIT=2314, VOL=SER=NUSYS0, DSN=KEDAK,
//_bDISP=(NEW,KEEP), SPACE=(TRK,300)
//G.FT17F001   DD   UNIT=2314, VOL=REF=*.FT14F001,
//_bDSN=*.FT14F001, DISP=OLD
//G.FT16F001   DD   UNIT=TAPE9, VOL=SER=901701, DSN=KERND,
//_bDISP=(,PASS), LABEL=(,SL),
//_bDCB=(RECFM=FB, LRECL=80, BLKSIZE=7200)
/*SETUP  DEVICE=TAPE9, ID=901701
//G.SYSIN   DD *
```

```
5  01717  01750  01701  01720  0
49  @H   1@  0010001  @H  H1@  0011001  @H  01@  0012001
    @H   2@  0010002  @HE  3@  0020003  @HE  4@  0020004
    @C  12@  0060012  @N   @  0070000  @O  16@  0080016
    @NA 23@  0110023  @AL 27@  0130027  @CR  @  0240000
    @CR 50@  0240050  @CR 52@  0240052  @CR 53@  0240053
    @CR 54@  0240054  @FE  @  0260000  @FE 54@  0260054
    @FE 56@  0260056  @FE 57@  0260057  @FE 58@  0260058
    @NI  @  0280000  @NI 58@  0280058  @NI 60@  0280060
    @NI 61@  0280061  @NI 62@  0280062  @NI 64@  0280064
    @MO  @  0420000  @MO 92@  0420092  @MO 94@  0420094
    @MO 95@  0420095  @MO 96@  0420096  @MO 97@  0420097
    @MO 98@  0420098  @MO100@  0420100  @U 235@  0920235
    @U 238@  0920238  @PU239@  0940239  @PU240@  0940240
    @PU241@  0940241  @PU242@  0940242  @CD  @  0480000
    'CL   UNC'  0170000  @CL 35@  0170035  @CL 37@  0170037
    'U 238IN1'  0922381  'U 238IN2'  0922382  'PB   EN3'  820000
    'U 238WC1'  0922383
69  @ISOT1@  14580  @ISOT2@  14590  @ISOT3@  14600  @PLNUE@  14570
    @CHICR@  14560  @RES  @  21520  @ST   @  21530  @STD  @  21540
    @STGF @  21550  @SGT  @  30010  @SGN  @  30020  @SGX  @  30030
    @SGI  @  30040  @SGIZ @  30050  @SG2N @  30160  @SG3N @  30170
    @SGF  @  30190  @SGIA @  30220  @SGI3A@  30230  @SG2NA@  30240
    @SG3NA@  30250  @SGA  @  30270  @SGIP @  30280  @SGNI @  30290
    @SGG  @  31020  @SGP  @  31030  @SGD  @  31040  @SGH3 @  31050
    @SGHE3@  31060  @SGALP@  31070  @SG2HE@  31080  @SGTR @  32010
    @ETA  @  32060  @ALPHA@  32070  @MUEL @  32510  @NUE  @  34520
    @NUEP @  34550  @CHIF @  34610  @CHIFD@  34620  @SGNL @  40021
    @SGNC @  40022  @SGIL @  40041  @SGIC @  40042  @SGILZ@  40051
    @SGICZ@  40052  @SGNIL@  40291  @SGNIC@  40292  @LEGNL@  44631
    @LEGNC@  44632  @LEGIL@  44641  @LEGIC@  44642  'LEGILZ'  44651
    'LEGICZ'  44652  @LGNIL@  44661  @LGNIC@  44662  @CHIFZ@  54610
    'CHIFDZ'  54620  @CHII @  50040  @CHI2N@  50160  'AASTATUS'  14510
    'RANGRES'  14511  'SGIZC '  30051  @CHI3N@  50170  @SEDF @  54523
    @SEDFF@  54613  @SEDFD@  54623  @SED2N@  50163  @SED3N@  50173
    @SEDIC@  50053
1  0  150972  1
@PU239@  @SGIZ @  2
81  @ADD  @  2  @NI   @  'AASTATUS'  1  1
 'DATA RE-EVALUATION IN COMPARISON WITH KFK 120 (1966)
 '1970 :   SGG    1 MEV - 15 MEV,
 '         SGALP, SGP, SG2N   THRESHOLD - 15 MEV,
 '         SGT, SGN, SGI, MUEL   10 MEV - 15 MEV.
13  @ADD  @  2  @NI   @  @ISOT1@  0  3  'ARG '  58.69  28.  0.
13  @ADD  @  2  @NI   @  @SGG  @  1  1  4.266E6  0.0016
 4.366E6  0.0015
11  @DROPS@  2  @FE   @  @SGT  @  1  1.35E6  2.72E6  3.807E6
18  @ADD  @  3  @PU239@  @SGIZ @  57000.  1  1  2.9E5
 0.145  4.E5  0.175  5.E5  0.191  5.5E5  0.197
16  @ADD  @  3  @PU239@  @SGIZ @  164000.  1  1  1.7E5  0.001
 1.8E5  0.002  2.E5  0.004
5  @DROPA@  1  'CL   UNC'
2  @ENDE @
'KEDA'  'BIBL'  'IOTH'  150972  45
```

V. Literature References

[1] D. Woll, KFK 880 (EANDC(E)-112 "U") Dec. 1968
[2] M.K. Drake, BNL 50274, Vol. I

VI. Appendix

Table 1 : Data types foreseen on KEDAK

(status 1972)

Name of data type K G S	Name as in ENDF/B?(1)	Name of data type on internal KEDAK	Further names	Arguments	Functional values
1 451 o	y	AASTATUS	-	-	1 (4) bibliographic information giving data types and energy regions of recent evaluations.
1 451 1	n	RANGRES	-	-	1. E_L - lower energy boundary of the region in which resolved resonance parameters 2. E_U - upper are given under data type " RES " 3. number of resolved resonances given by " RES " 4. flag which indicates whether resolved resonance parameters should preferable be taken for group constant calculations or pointwise given cross section values. It may have the following values. 2. - cross section values should be taken 1. - resolved resonance parameters o. - no preference can be recommended
1 458 o	n	ISØT1	-	-	1. Atomic weight (A) 2. Atomic number (Z) 3. Nuclear spin of ground state (I)
1 459 o	n	ISØT 2	-	-	1. ~~λ~~ $\sqrt{E} = \hbar / \sqrt{2m_n} \cdot \frac{A+1}{A}$ reduced neutron wave length $\left[eV^{1/2} b^{1/2} \right]$ 2. R = nuclear radius $\left[b^{1/2} \right]$ 3. E_B = binding energy of the last neutron in compound nucleus
1 46o o	n	ISØT 3	-	Isotope atomic weight	Isotopic abundance (%)
1 457 o	n	PLNUE	-	-	1. ν_o 2. ν_1 3. ν_2 4. ν_3 where $\nu = \sum_{i=o}^{3} \nu_i E^i$ = average total number of fission neutrons

Name of data type K G S	Name as in ENDF/B?(1)	Name of data type on internal KEDAK	Further names	Arguments	Functional values
1 456 o	n	CHICR	-	1. Neutron incident energy	1. A 2. B 3. C } Parameters of the Cranberg fission spectrum
2 152 o	n	RES	-	1. Resonance energy 2. Neutron orbital angular momentum (l) 3. Compound nucleus spin (J)	1. $g_J = (2J+1)/(2(2I+1))$ 2. total half width Γ 3. neutron half width Γ_n 4. capture half width Γ_γ 5. fission half width Γ_f 6. (n,p)-half width Γ_p 7. (n,α)-half width Γ_α 8. (n,n')-half width $\Gamma_{n'}$
2 15 30	n	ST	-	1. L 2. J	1. average capture width $\overline{\Gamma_\gamma}$ 2. average level spacing $\overline{D}$ 3. average reduced neutron width $\overline{\Gamma_n^{(o)}}$ 4. strength function $\overline{\Gamma_n(o)}/\overline{D}$ 5. number of exit channels in fission ν_f 6. number of exit channels in neutron elastic scattering (ν_n)
2 154 o	n	STD	-	-	1. average observed level spacing 2. a 3. $2\sigma^2$ } parameters of the statistical theory

Name of data type K G S	Name as in ENDF/B?(1)	Name of data type on internal KEDAK	Further names	Arguments	Functional values
2 155 o	n	STGF	-	1. neutron incident energy 2. l 3. j	1. number of exit channels in fission ν_f 2. average fission width $\overline{\Gamma_f}$ for the number of exit channels ν_f 3. average capture width $\overline{\Gamma_\gamma}$ 4. average neutron width $\overline{\Gamma_n}$ 5. S_f 6. S_γ 7. R_f 8. R_γ } statistical fluctuation factors(3)
3 oo1 o	y	SGT	-	neutron incident energy	total cross section
3 oo2 o	y	SGN	-	"	elastic scattering cross section
3 oo3 o	y	SGX	-	"	non-elastic cross section
3 oo4 o	y	SGI	-	"	total inelastic cross section
3 oo5 o	n	SGIZ	E_i	"	inelastic cross section for excitation of rest nucleus level E_i
3 oo5 1	n	SGIZC	-	"	inelastic scattering cross section to the continuum
3 o16 o	y	SG2N	-	"	cross section for the (n,2n)-process
3 o17 o	y	SG3N	-	"	cross section for the (n,3n)-process
3 o19 o	y	SGF	-	"	fission cross section
3 o22 o	y	SGIA	-	"	cross section for the (n,n'α)-process
3 o23 o	y	SGI3A	-	"	" " " " (n,n'3α)- "
3 o24 o	y	SG2NA	-	"	" " " " (n,2nα)- "
3 o25 o	y	SG3NA	-	"	" " " " (n,3nα)- "

Name of data type K G S	Name as in ENDF/B?(1)	Name of data type on internal KEDAK	Further names	Arguments	Functional values
3 027 o	y	SGA	-	neutron incident energy	absorption cross section
3 028 o	y	SGIP	-	"	cross section for the (n,n'p)-process
3 029 o	y	SGNI	-	"	" " " " sum of σ_n and $\sigma_{n'}$
3 102 o	y	SGG	-	"	" " " " (n,γ) -process
3 103 o	y	SGP	-	"	" " " " (n,p) - "
3 104 o	y	SGD	-	"	" " " " (n,d) - "
3 105 o	y	SGH3	-	"	" " " " (n,H^3) - "
3 106 o	y	SGHE3	-	"	" " " " (n,He^3)- "
3 107 o	y	SGALP	-	"	" " " " (n,α) - "
3 108 o	y	SG2HE	-	"	" " " " $(n,2\alpha)$ - "
3 201 o	n	SGTR	-	"	transport cross section
3 206 o	n	ETA	-	"	average number of fission neutrons per neutron absorption
3 207 o	n	ALPHA	-	"	ratio of capture to fission cross section
3 251 o	y	MUEL	-	"	average cosine of the elastic scattering angle in the laboratory system $\cos\theta_L = \overline{\mu_L}$
3 452 o	y	NUE	-	"	average number of fission neutrons
3 455 o	n	NUEP	-	"	average number of prompt fission neutrons
3 461 o	n	CHIF	-	neutron outgoing energy	energy spectrum of prompt fission neutrons (thermal fission)
3 462 o	n	CHIFD	-	"	energy spectrum of delayed fission neutrons (thermal fission)

Name of data type K G S	Name as in ENDF/B?(1)	Name of data type on internal KEDAK	Further names	Arguments	Functional values
4 oo2 1	n	SGNL	E_o (2)	cosine of scattering angle	differential elastic scattering cross section at the neutron incident energy E_o in the laboratory system
4 oo2 2	n	SGNC	E_o (2)	"	differential elastic scattering cross section at the neutron incident energy E_o in the center-of-mass system
4 oo4 1	n	SGIL	E_o	"	differential inelastic scattering cross section at the neutron incident energy E_o in the laboratory system
4 oo4 2	n	SGIC	E_o	"	differential inelastic scattering cross section at the neutron incident energy E_o in the center-of-mass system
4 oo5 1	n	SGILZ	1.E_i 2.E_o	"	differential inelastic scattering cross section for excitation of the rest nucleus level E_i at the neutron incident energy E_o in the laboratory system
4 oo5 2	n	SGICZ	1.E_i 2.E_o	"	differential inelasic cross section for excitation of the rest nucleus level E_i at the neutron incident energy E_o in the center-of-mass system
4 o29 1	n	SGNIL	1.E_2 2.E_o	"	differential cross section for elastic and inelastic scattering at the neutron incident energy E_o to neutron outgoing energies between E_o and E_2 in the laboratory system
4 o29 2	n	SGNIC	1.E_2 2.E_o	"	differential cross section for elastic and inelastic scattering at the neutron incident energy E_o to neutron outgoing energies between E_o and E_2 in the center-of-mass system
4 463 1	n	LEGNL	1.E_o 2.order L_m	L	coefficient f_L in the Legendre-polynomial expansion of the differential elastic scattering cross section

Name of data type K G S	Name as in ENDF/? (1)	Name of data type on internal KEDAK	Further names	Arguments	Functional values
					$\sigma_n(\theta)=\frac{\sigma_n}{4\pi}\sum_{L=o}^{L_m}(2L+1)\,f_L(E)\,P_L(\cos\theta)$ in the laboratory system
4 463 2	n	LEGNC	1. E_o 2. order L_m	L	coefficent f_L in the Legendre-polynomial expansion of the differential elastic scattering cross section $\sigma_n(\theta)=\frac{\sigma_n}{4\pi}\sum_{L=o}^{L_m}(2L+1)\,f_L(E)\,P_L(\cos\theta)$ in the center-of-mass system
4 464 1	n	LEGIL	1. E_o 2. order L_m	L	coefficient f_L' in the Legendre-polynomial expansion of the differential inelastic scattering cross section $\sigma_n,(\theta)=\frac{\sigma_n'}{4\pi}\sum_{L=o}^{L_m}(2L+1)\,f_L'(E)\,P_L(\cos\theta)$ in the laboratory system
4 464 2	n	LEGIC	1. E_o 2. order L_m	L	coefficient f_L' in the Legendre-polynomial expansion of the differential inelastic scattering cross section $\sigma_n,(\theta)=\frac{\sigma_n'}{4\pi}\sum_{L=o}^{L_m}(2L+1)\,f_L'(E)\,P_L(\cos\theta)$ in the center-of-mass system
4 465 1	n	LEGILZ	1. E_i 2. E_o 3. order L_m	L	coefficient f_L^i in the Legendre-polynomial expansion of the differential inelastic cross section for excitation of the rest nucleus level E_i

Name of data type K G S	Name as in ENDF/?(1)	Name of data type of internal KEDAK	Further names	Arguments	Functional values
					$\sigma_{n'}^{E_i}(\theta) = \frac{\sigma_{n'}}{4\pi} \sum_{L=o}^{L_m} (2L+1)\, f_L^i(E)\, P_L(\cos\theta)$ in the laboratory system
4 465 2	n	LEGICZ	1. E_i 2. E_o 3. order L_m	L	coefficient f_L^i in the Legendre-polynomial expansion of the differential inelastic cross section for excitation of the rest nucleus level E_i
					$\sigma_{n'}^{E_i}(\theta) = \frac{\sigma_{n'}}{4\pi} \sum_{L=o}^{L_m} (2L+1)\, f_L^i(E)\, P_L(\cos\theta)$ in the center-of-mass system
4 466 1	n	LGNIL	1. E_2 2. E_o 3. order L_m	L	coefficient f_L^{o2} in the Legendre-polynomial expansion of the differential cross section for elastic and inelastic scattering at the neutron incident energy E_o to neutron outgoing energies between E_o and E_2
					$\sigma_{n+n'}^{o2}(\theta) = \frac{\sigma^{o2}_{n+n'}}{4\pi} \sum_{L=o}^{L_m} (2L+1)\, f_L^{o2}(E)\, P_L(\cos\theta)$ in the laboratory system
4 466 2	n	LGNIC	1. E_2 2. E_o 3. order L_m	L	coefficient f_L^{o2} in the Legendre-polynomial expansion of the differential cross section for elastic and inelastic scattering at the neutron incident energy E_o and E_2
					$\sigma_{n+n'}^{o2}(\theta) = \frac{\sigma^{o2}_{n+n'}}{4\pi} \sum_{L=o}^{L_m} (2L+1)\, f_L^{o2}(E)\, P_L(\cos\theta)$ in the center-of-mass system

Name of data type K G S	Name as in ENDF/? (1)	Name of data type of internal KEDAK	Further names	Arguments	Functional values
5 461 o	n	CHIFZ	E_o	neutron outgoing energy	energy spectrum of prompt fission neutrons at the neutron incident energy E_o
5 462 o	n	CHIFDZ	E_o	"	energy spectrum of delayed fission neutrons at the neutron incident energy E_o
5 oo4 o	y	CHII	E_o	"	energy spectrum of inelastically scattered neutrons at the neutron incident energy E_o
5 o16 o	y	CHI2N	E_o	"	1.)2.) energy spectrum of the two neutrons emitted in the (n,2n) process at the neutron incident energy E_o
5 o17 o	y	CHI3N	E_o	"	1.)2.)3.) energy spectrum of the three neutrons emitted in the (n,3n) process at the neutron incident energy E_o

Name of data type K G S	Name as in ENDF/?[1]	Name of data type of internal KEDAK	Further names	Arguments	Functional values
					parametric representation of energy spectra at incident neutron energy E_o
5 oo5 3	n	SEDIC	E_o	K-identification number for the model used for description :	of neutrons inelastically scattered to a continuum of levels
5 o16 3	y	SED2N	"		of the two neutrons emitted by the (n,2n) process
5 o17 3	y	SED3N	"	K = 1 evaporation spectrum	of the neutrons emitted by the (n,3n) process
5 452 3	n	SEDF	"	K = 2 Maxwellian spectrum	of fission neutrons
5 461 3	n	SEDFP	"	K = 3 Watt-Cranberg spectrum	of prompt fission neutrons
5 462 3	n	SEDFD	"	K = 4 Excitation of discrete levels	of delayed fission neutrons
					3 functional values :[5] 1. p - fraction of the spectrum of type K to the total energy distribution 2. θ (nuclear temperature) - for K = 1,2 a (spectrum parameter) - for K = 3 EC (level excitation energy) - for K = 4 3. K - upper limit for the final neutron energy $0 \leq E' \leq E - U$ - for K = 1,2 or b (spectrum parameter) - for K = 3 or o - for K = 4

(1) K always corresponds to the ENDF/B format. If also G corresponds to the ENDF/B format, then the second column contains " yes ", otherwise " no ".

(2) E_o for this and all pertinent further data types in the laboratory system. This is also true for E_2.

(3) $$S_f = \frac{\overline{|\gamma}}{\overline{|n}\ \overline{|f}} \left\langle \frac{\overline{|n}\ \overline{|f}}{|} \right\rangle ; \quad S_\gamma = \frac{\overline{|\gamma}}{\overline{|n}} \left\langle \frac{\overline{|n}}{|} \right\rangle ; \quad R_f = \frac{\overline{|\gamma}}{\overline{|n^2}\ \overline{|f}} \left\langle \frac{\overline{|n^2}\ \overline{|f}}{|} \right\rangle ; \quad R_\gamma = \frac{\overline{|\gamma}}{\overline{|n^2}} \left\langle \frac{\overline{|n^2}}{|} \right\rangle$$

(4) The data items of AASTATUS are only formally divided into argument and functional value. They contain the indicated text in successive order.

(5) These formats were chosen in accordance with ENDF/B as described in ref. [2].