KERNFORSCHUNGSZENTRUM

KARLSRUHE

April 1977 KFK 2387/III

Institut für Neutronenphysik und Reaktortechnik
Projekt Schneller Brüter

The KEDAK Program Compendium
Part III
KEDAK Data Retrieval

Compiled by: E. Stein
with contributions from
B. Krieg, I. Langner, R. Meyer, E. Stein

GESELLSCHAFT
FÜR
KERNFORSCHUNG M.B.H.

KARLSRUHE

GESELLSCHAFT FÜR KERNFORSCHUNG M. B. H.
KARLSRUHE

KERNFORSCHUNGSZENTRUM KARLSRUHE

KFK 2387/III

Institut für Neutronenphysik und Reaktortechnik
Projekt Schneller Brüter

The KEDAK Program Compendium

Part III

KEDAK Data Retrieval

Compiled by

E. Stein

with contributions from

B.Krieg, I.Langner, R.Meyer*, E.Stein

* Present address: Software AG, Darmstadt

Gesellschaft für Kernforschung mbH., Karlsruhe

Abstract

All data retrieval packages available for the direct access representation of KEDAK will be described in this part of the Compendium. Both the basic access programs, which retrieve one data item per access, and the user oriented programs, which retrieve, by using the basic programs, a set of data items, will be presented. Besides the detailed description also a general introduction to the KEDAK data retrieval will be given.

Das KEDAK Programm Compendium
Teil III
Daten Retrieval von KEDAK

Zusammenfassung

Alle verfügbaren Daten Retrieval Pakete für die Darstellung von KEDAK, welche wahlfreien Zugriff erlauben, sind in diesem Teil des Kompendiums beschrieben. Sowohl die Grundzugriffs-Programme, welche pro Zugriff einen Datenpunkt anliefern, als auch die benutzerorientierten Programme, welche unter Benutzung der Grund-Programme einen Satz von Datenpunkten bereitstellen, werden dargestellt. Neben der detaillierten Beschreibung erfolgt auch eine allgemeine Einführung in das Daten Retrieval von KEDAK.

Acknowledgement

The authors of this report thank Mrs. K. Mayer for typing this report with great carefulness and patience. Besides typing Mrs. K. Mayer produced out of the figures, which have been given as sketches, reproducable graphical representation.

Contents

Page

1. Introduction in the data retrieval from the evaluated nuclear data library KEDAK. III-1
   E. Stein

2. IDFPAC-LDFPAC. Two retrieval packages for the evaluated nuclear data library KEDAK for use in applications programs. III-9
   I. Langner, R. Meyer

3. NDF. A retrieval program for the evaluated nuclear data library KEDAK for use in applications programs. III-141
   B. Krieg

4. RETPAC. A user oriented retrieval package for use with the evaluated nuclear data library KEDAK. III-157
   R. Meyer

Zum Druck am: 23.3.1977

Section 1

Introduction in the data retrieval from the evaluated nuclear data library KEDAK

E. Stein

The information, which is contained in KEDAK, is stored in units of data items. One access to KEDAK retrieves one and only one data item. Examples for data items are:

- one energy value together with the attached cross section value,
- a resonance energy value, a neutron orbital angular momentum, a compound nucleus spin together with the attached statistical factor $g_J = \frac{2\cdot J+1}{2\cdot(2\cdot I+1)}$, and the resolved half widths Γ, Γ_n, Γ_γ, Γ_f, Γ_p, Γ_α and $\Gamma_{n'}$,
- the atomic weight, the atomic number, and the nuclear spin of the ground state.

The data items for a specified material and a specified neutron reaction type (or in general data type for non neutron reaction data) are collected in groups. Such a group is identified by a material name and a neutron reaction type name. Examples for names are:

- U 238, SGT	total neutron reaction of Uranium 238
- Pu 239, RES	resolved resonance parameter set for Plutonium 239
- Pu 239, ISOT1	atomic weight, atomic number, and nuclear spin of ground state for Plutonium 239.

The first two data types (SGT and RES) normally contain several data items: in the case of SGT one for each energy value. The datatype ISOT1 contains only one data item: the three values given above.

There are neutron reactions,which are not only identified by the material and reaction type name, but also by so called further or additional names. Examples for reaction types with further names are (the material name is also listed):

- U 238, SGIZ

  first name: U 238 meaning Uranium 238

  second name: SGIZ meaning inelastic level cross section

  first further name (or third name):
  real value meaning the level energy.

  (SGIZ has <u>one</u> further name.)

- PU 239 , LEGNC

  first name: PU 239 meaning Plutonium 239

  second name: LEGNC meaning the data type, which contains the Legendre coefficients of the expansion of the differential elastic scattering cross section

  first further name (or third name):
  real value meaning the energy E_o.

  second further name (or fourth name):
  real value meaning the order of the Legendre polynom

  (LEGNC has <u>two</u> further names.)

In the case of LEGNC a data item consists of the value ℓ and the attached coefficient f_ℓ of the Legendre polynomial expansion.

As described, the data on KEDAK are identified by a material name (first name) and a neutron reaction type name, resp. data type name (second name). Some reactions require additional names (third name, fourth name,...), which are always real numbers. Further examples for data types can be found in part II of this Compendium[+].

The KEDAK library can be represented in two modes:

(1) a direct access form

(2) a (sequential) card image form

[+] KFK 2387/II

Because the direct access form is much more convenient for random access, which is normally done in data retrieval, all codes, which are described in the following, presume the KEDAK direct access representation. But it is projected to develop retrieval codes for the card image form of KEDAK, which will have the same argument lists, as those subroutines, which are described in section 2.

The KEDAK direct access representation contains conversion tables for the attachment of alphamerical names and internally used numerical names. E.g. the alphamerical material name U 238 will be replaced for program internal use by the numerical name 0920238. In a similar way the alphamerical data type (neutron reaction) name SGT (total cross section) will be replaced by the program by the numerical name 30010. Additional examples for numerical names can be found in part II of this Compendium[+].

Mainly two methods are existing for the retrieval of data from KEDAK. One is called LDF-mode and the other one IDF-mode. In the LDF-mode alphamerical material and data type names are given by the user, and the program performs the translation in the internally used numerical material and data type names. In the IDF-mode the user himself prepares the internally used numerical names. Although retrieval, when using the IDF-mode, has a shorter searching time, because no conversion of alphamerical names must be done, we recommend to use only the LDF-mode. The advantage of a shorter searching time is negligible small compared with the expense of using numerical names, which must be searched in most cases in written tables.

Besides the LDF and IDF-mode, the NDF mode is existing. This mode has been used initially to retrieve data from the direct access form of the KEDAK library. When using the NDF mode, also alphamerical names, as in the LDF mode, must be given by the user. Yet the argument lists of the NDF-routines differ from those, which are used in the LDF mode routines. A description of the NDF mode, which should not be used in the futute, is included here, because some older programs are using this mode.

[+] KFK 2387/II

Each retrieval mode - LDF, IDF and also NDF - performs the same basic tasks. Due to this, all the subroutines or subroutine packages, which are realizing a specified retrieval mode, have entries respectively subroutines, which are performing these tasks. These are:

- OPN entry : the KEDAK library is opened, the first three words of the library, which must be KEDA BIBL IOTH, are tested, the conversion tables and address tables are read, and the data of the last change of the library is returned. An OPN call must be made once for each KEDAK-library data set, which is to be used in the program.

- LOC entry : tests if the desired information, marked by material and data type name[1], can be found on the library. If not found, the return code NR in the argument list is set to zero. In the other case, if the data can be located in the library, the return code NR is set to one and the first data item is retrieved. If the data type has further names, those set of further names is searched, which is logically the first one[2] having stored further name values, which are all greater or equal to the corresponding further names given by the user in the argument list. The further names in the argument list are set to the values found in the library and the first data item of this subgroup (belonging to this set of further names) is returned. The return code NR is set to one in the case of further names, if there exists a set of stored further names with values, which are all greater or equal compared with that one given by the user. This statement is only valid for the routines,

[1] For the routines, which are described in section 3, also the number of names must be specified.

[2] The sets of further names are arranged in increasing values, with the last name changing most quickly and the first name changing most slowly.

which are described in section 2. The routines in section 3 will set the return code NR to a value of one, only if the values of a set of stored further names are equal to the corresponding values, which are given by the user. Due to this reason a first LOC-call will result, in the case of further names and of the routines of section 3, normally in a NR=0 setting. Yet in this call the further names are set to those stored - the logically first one - combination of further names with values greater than or equal to the corresponding one given by the user. By this a second LOC-call may result in a NR=1 setting. Naturally, if no data of the desired type exist, or if no combination of further names can be found, which have values greater than or at least equal to the given ones, a second LOC-call will also return NR=0.

- NXT entry: retrieves the next data item. The data item following that one, which has been retrieved in a previous NXT- or LOC-call, is retrieved. If the process is successful, the return code NR is set to one. If the call fails, because the last data item of the specified data type has been retrieved in the previous NXT-call (or LOC-call for types with only one item), NR is set to zero. In the case of further names, the next combination of further names - if existing - and the first data item belonging to this combination is returned together with NR=0, if the last data item, belonging to a combination of further names, has been retrieved in a previous call. A following NXT-call will retrieve the second data item. But we recommend to use after NR=0 in a NXT-call always a LOC-call, which may use the combination of further names, which has been retrieved by the NR=0 NXT-call.

Different programs - respectively program packages - have been produced for the different retrieval modes. All these programs - respectively packages - contain in every case three entry - respectively subroutine - names, to perform the three tasks described above. The first three characters of the names relate to the retrieval mode, the last three characters to the task (OPN, LOC, or NXT), which is to be performed. When using this rule, the following names are possible:

NDF mode	IDF mode	LDF mode
NDFOPN	IDFOPN	LDFOPN
NDFLOC	IDFLOC	LDFLOC
NDFNXT	IDFNXT	LDFNXT

Three program packages exist for the LDF mode and two for the IDF mode. We recommend to use the packages described in section 2, because there are some advantages when using these programs compared with the programs of section 3. These advantages are:

- the number of names must not be given by the user
- only one LOC-call in the case of further names
- a splitting up of the KEDAK library in up to five different datasets is allowed
- the packages of section 2 consist of a set of subroutines, which can be overlayed
- there are some additional entries in the package, which can be used to perform special tasks (see section 2).

Because the programs, which are described in section 2, use the same argument lists and the same subroutine resp. entry names, as those, which are described in section 3, the new programs of section 2 are compatible with the older programs of section 3. That means, the older programs can be replaced by the new ones.

Besides the description of the LDF and IDF retrieval routines, section 3 contains also the description of the NDF routine. This is done only due to historical reasons and because this mode is used in older programs. The NDF mode will not be supported in the future.

Section 2 contains two program packages, called IDFPAC and LDFPAC. Only these packages should be used in new programs. The IDFPAC contains both programs for the IDF retrieval mode and programs for the LDF retrieval mode. The LDFPAC only performs the LDF retrieval.

Until now, we have restricted our considerations to programs, which make a direct access to the KEDAK library and retrieve one data item per access. Yet users of KEDAK are normally not only interested in one data item, but e.g. in the cross sections in a given energy range. To facilitate this task, so called "user oriented retrieval subroutines" have been written. This set of subroutines is called RETPAC. It is described in section 4. Subroutines belonging to RETPAC use the basic retrieval routines, which are described in section 2.

At last some explanation is given with respect to the numeration used in the following. Because section 2 has been written as an independant report, the number of this section is not contained in the local numeration of this section. E.g. the local number 3.1.1 of section 2 must be prefixed with section number 2. This results in 2.3.1.1. Because this section belongs to part III, the correct numeration of 3.1.1 in section 2 in chapter III is: III.2.3.1.1. This numeration is used only, if references outside of a section are made. Within section 2 the local numeration without the section and chapter number is used only.

Section 2

IDFPAC - LDFPAC

Two retrieval packages for the
Evaluated Nuclear Data Library KEDAK
for use in application programs

by

I. Langner
R. Meyer*

*Present address: Software AG, Darmstadt

Contents

Page

1. Introduction III-14

1.1 Purpose of this report III-14

1.2 Purpose of IDFPAC and LDFPAC III-14

1.3 Summary of the structure of the KEDAK library III-16

1.3.1 Categories of data III-16

1.3.2 Arrangement of data III-17

1.4 Application of IDFPAC and LDFPAC III-19

1.5 Entry points III-19

2. User's guide III-22

2.1 Purpose and use of LDFOPN, LDFLOC and LDFNXT III-22

2.1.1 RETXS - an example for the use of LDFLOC and LDFNXT III-25

2.2 Purpose and use of LDFITN, LDFNAM III-29

2.2.1 FILLTP - an example for the use of LDFITN III-30

2.2.2 Job control statements for the printing program III-32

2.2.3 The program to print the contents of a KEDAK-file as an example for use of LDFOPN, LDFITN, LDFNAM III-33

2.3 Storage requirements and availability III-36

2.4 The call scheme for IDFPAC and LDFPAC III-36

2.5 Possible overlay structures for IDFPAC and LDFPAC III-39

2.6 Unresolved reference DEFI III-40

2.7 Flow chart III-41

3. Programmer's guide - detailed description III-42

3.1 Detailed description of the labeled common blocks III-42

3.1.1 LDFMT - to retain the material address-table and type address-table for the KEDAK-file just processed III-42

3.1.2 LDFTT - to retain the type address-table for the material just processed III-43

3.1.3 LDFRQ - contains the names and information necessary at the LDFLOC, LDFNXT calls for the requested material III-44

3.1.4 LDFRE - to store the last record read from the KEDAK-file III-46

3.1.5 LDFIL - to retain information about the KEDAK-file just processed III-46

Page

3.1.6 LDFTC - to retain the type conversion table of the KEDAK-file just processed III-47

3.1.7 LDFRC - to provide information for LDFERR, the error diagnostic message routine III-47

3.2 Detailed description of the user called subroutines LDFOPN, LDFLOC, LDFNXT, LDFITN, LDFNAM III-48

3.2.1 LDFOPN - to define the KEDAK-file and read the first three tables (see Appendix 1) III-48

ENTRIES: LDFROP - to reread the tables of the current file III-49

LDFALF - to load the tables of an alternative file III-49

3.2.2 LDFLOC - to locate data of requested type for alphameric names III-51

ENTRIES: IDFLOC - to locate data for numeric names III-52

LDFTTO - to provide the type address-table III-52

LDFRQO - to provide the contents of the LDFRQ-common III-52

3.2.3 LDFNXT - to retrieve the next data item for alphameric names III-54

ENTRY: IDFNXT - to retrieve the next data item for numeric names III-54

3.2.4 LDFITN - to provide some information from the type address table III-54

3.2.5 LDFNAM - to provide information from the address table of further names III-56

3.2.6 LDFERR - to print error messages III-58

3.3 Detailed description of the auxiliary subroutines III-59

3.3.1 LDFTB1 - to load information from the material conversion table, the type conversion table and the material address table into the LDFTC and LDFMT common blocks III-59

3.3.2 LDFCTB - to retrieve the conversion tables III-60

3.3.3 LDFMAC - to convert the numeric names in the material address table to alphameric names III-60

3.3.4 LDFMAT - to retrieve the material address table III-61

3.3.5 LDFTAT - to retrieve the type address table for a specified material III-63

3.3.6 LDFTYP - to retrieve pointers to the type address table for a specified material III-64

3.3.7 LDFFNA - to search for data types with further names and retrieve the pointers III-64

3.3.8 LDFREC - to read one record from the KEDAK-file III-65

3.3.9 LDFTYC - to retrieve pointers for a specified data type III-66

3.3.10 LDFDPN - to provide the number of data points III-66

Page

4. References III- 68

5. Source lists oi IDFPAC III- 70

6. Description of LDFPAC III- 88

7. Source lists of LDFPAC III- 90

Appendix 1: Material and type name conversion table, the table of available isotopes and the types available for each isotope listed by the program to print the contents of a KEDAK-file. III-106

Appendix 2: Lay-out of logical records on the KEDAK-file III-133

1. Introduction

1.1 Purpose of this report

This report is intended as documentation of the two subroutine packages IDFPAC and LDFPAC for retrieval of information from the Evaluated Nuclear Data Library KEDAK (Kerndatenbibliothek Karlsruhe).

This documentation gives all information needed to use the capability of these packages to their full extent. The maintenance information needed to support the packages is also covered.

1.2 Purpose of IDFPAC and LDFPAC

A data library is a collection of an usually large amount of data intended for permanent use in specific application areas. Both size and frequent usage necessitate a rather sophisticated organization of the data for quick insertion, replacement or deletion in case of updates and for rapid retrieval in case of usage.

Program packages designed for retrieval or updating of information within the data library therefore may become quite complicated and require a detailed knowledge of the organization of the data library. To relieve the user of the task of writing this type of programs individually, subroutine packages are usually made available to the data users, which perform these tasks and form a sort of "interface" between his application program and the data library. These packages are confined to retrieval of the information. Fig. 1 shows the interfacing roll in case of IDFPAC and LDFPAC.

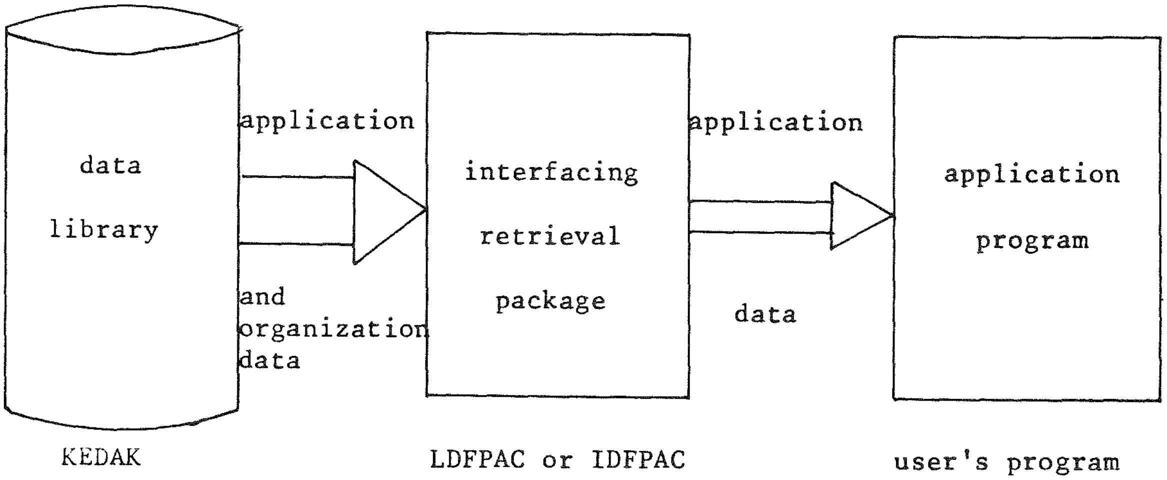


Fig. 1: Interfacing retrieval programs

The application programs call subroutines within the retrieval packages and tell them simply what kind of data are required. The subroutine packages rapidly locate the desired information within the library and transfer it into user supplied storage areas (e.g. array locations the address of which is given by the user).

IDFPAC and LDFPAC are two retrieval packages for the Evaluated Nuclear Data Library KEDAK. Each of them consists of a set of FORTRAN subroutines that perform all operations necessary to locate and read a given set of data from the KEDAK library. The user, who wants to retrieve information from KEDAK, has to call entry points within these packages in a standard manner, which is described later in this report. A subroutine call specifies by means of its arguments,which information should be read from the library and where to deposit it. The user need not care about the structure and actual location of data. Fig. 2 describes the flow of information for the retrieval process.

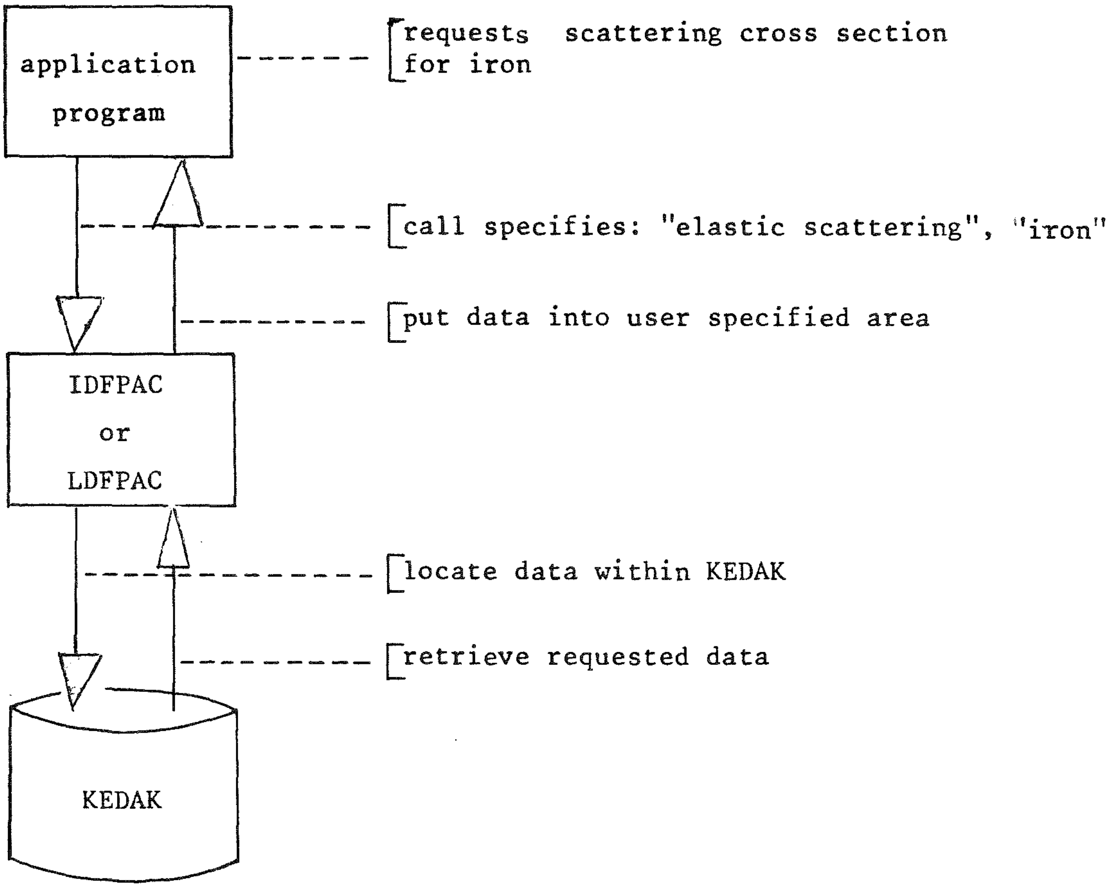

application program
requests scattering cross section for iron
call specifies: "elastic scattering", "iron"
put data into user specified area
IDFPAC or LDFPAC
locate data within KEDAK
retrieve requested data
KEDAK

Fig. 2: Flow of information when using the retrieval packages IDFPAC / LDFPAC

The subsequent chapter will give a short summary of the physical and logical structure of KEDAK to the interested user. For a complete and detailed presentation see reference 1.

1.3 Summary of the structure of the KEDAK library

The KEDAK library is used in Karlsruhe in a direct access form and only this format will be presented here.

1.3.1 Categories of data

The nuclear information is categorized by the name of the material to which it relates (e.g. Fe, Fe-56 and similar) and by the kind (or type) of data it presents, e.g. total cross section, elastic scattering angular distributions, resonance parameters, etc. To each material and data type two unique designations are assigned: one being an integer number called "numeric name" and the other one being an eight character "alphameric name". Alphameric names have been designated for the convenience of the user, numeric names are used within the architecture of the library and have been chosen in consistency to other evaluated nuclear data libraries (ENDF/B, UKNDL). To the user they appear fully synonymous and he may use either name he wishes. The use of alphameric names is easier, because they are commonly used abbreviations, which are e.g. containing the chemical symbols in the case of material names. The material name, which is the major key to the information, is called "first name" . The data type is called "second name" and may require further (third, fourth,etc.)names for unique identification. For example the inelastic excitation cross section (second name = inelastic excitation) implies additionally the specification of the excitation level energy (third name). In this case the level energy is called "further name" of this data type.

In most cases the quantities stored in the library are functions of one or several "arguments", e.g. the total neutron cross section for iron is a function of incident neutron energy, the resonance parameters are functions of resonance energy and neutron angular momentum, etc.

Thus "arguments" are the independent quantities at which the values of the data types are stored in the library and a data type may depend (as implied above) on one or several arguments. The numerical values of the nuclear data type corresponding to specific values of the arguments are accordingly termed "functional values" and a data type may specify one or several functional values.

1.3.2 Arrangement of data

The KEDAK library consists of a declaration part and a data part.

The purpose of the declaration part is to provide necessary addressing information to access the data on the library by a hierarchy of pointers (see Fig. 3). The pointers are stored in the "addresses". Each address consists of two 4-byte words: the first is the record number, the second the number of the desired word in the record where the information begins.

The declaration part starts with

the library identification - 3 words alphameric text,
the creation date - the date of the last update run,

followed by:

the number of isotopes in the material conversion table,
address of the material name conversion table, (see Appendix 1)

the number of types in the type name conversion table,
address of the type name conversion table

the number of isotopes available on the KEDAK library,
address of the available isotope list = starting point of the material address table,

and contains:

the material conversion table, (see Appendix 1)

the type conversion table, (see Appendix 1)

the material address table
contains for each isotope available:
the numeric material (isotope) name,
the number of types available for the isotope,
address of the type address table,

The data part consists for each isotope of the following blocks:

the type addresstable for the isotope:
the numeric type name,
NFN - number of further names for this type,
NARG - number of arguments,
NFV - number of function values,

if the number of further names is equal zero:
NUM - number of data points (items),
the address of the data array,

if the number of further names is not equal zero:
NUM - number of name combinations (data types),
the address of the name combination list (further name addresstable)

Repeated NUM-times for all combinations of further names:

the further names address table:
the further names (floating point notation),
NUM-number of data points,
the address of the data array,

The data array: the argument(s), the function value(s),	a data item, NUM times repeated for ascending values of the argument(s)

The data arrays are given for each type of the isotope available on KEDAK.

For detailed information see reference 1.

1.4 Applications of IDFPAC / LDFPAC

IDFPAC retrieves data using numeric or alphameric names optionally. The LDFPAC retrieval package performs the same functions as IDFPAC but for alphameric names only. IDFPAC is used in the SCORE-Version-III-K-1972 (see reference 2) ,LDFPAC is used in CALCUL (see reference 6), PLKFG (see reference 7), COPEND (see reference 8), and other programs. Remember that both packages do apply to the direct access representation of KEDAK used in Karlsruhe and mentioned in 1.3.

1.5 Entry points

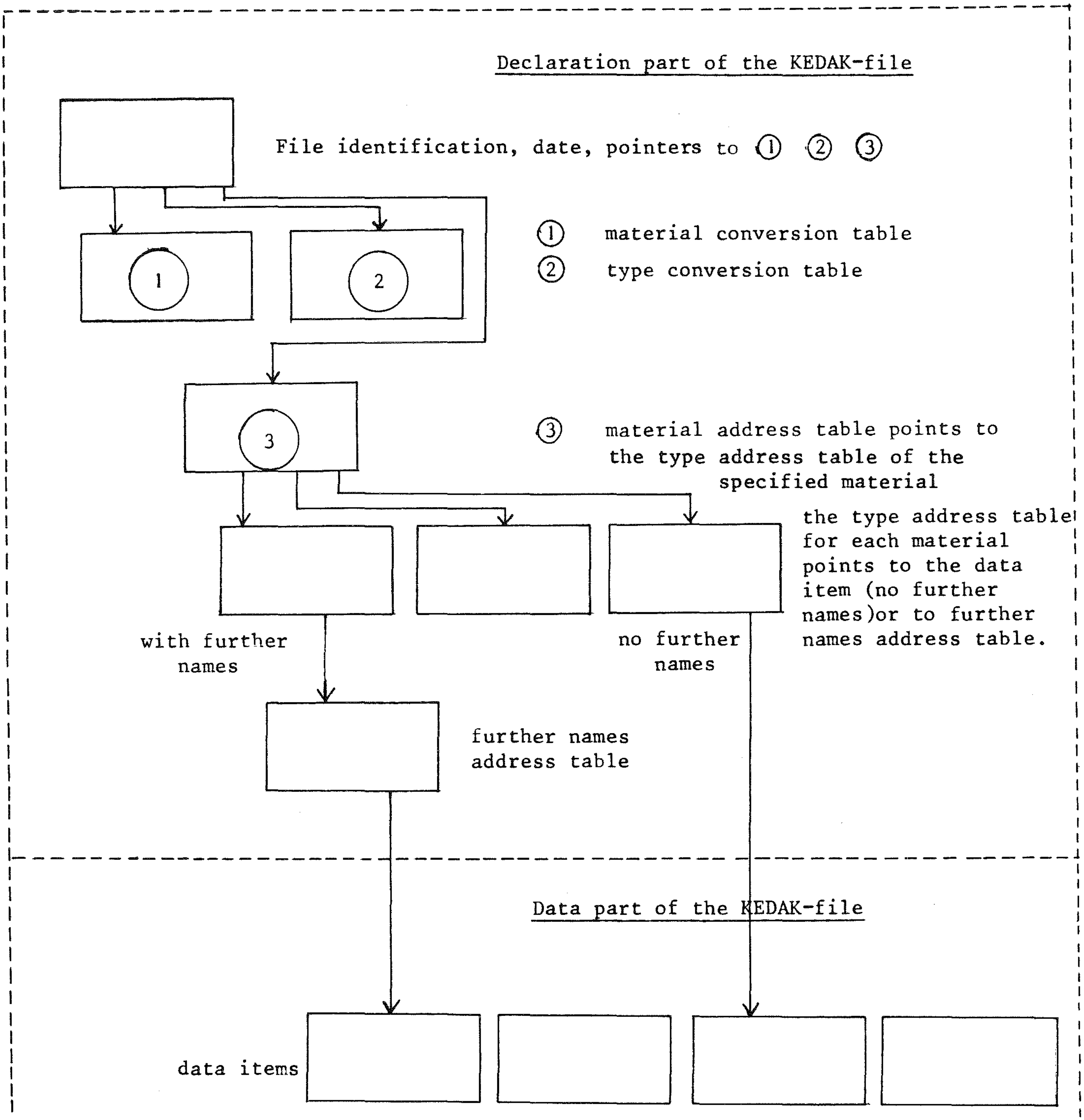
LDFOPN, LDFLOC and LDFNXT are the subroutines for information retrieval, which are usually called by the user's (application) program. LDFITN and LDFNAM are called by the user for special purposes. The subroutines LDFTYP, LDFTYC, LDFNNA, LDFTAT, LDFMAC, LDFCTB, LDFMAT, LDFREC, LDFERR, LDFDPN are auxiliary subroutines within the retrieval packages and are not called by the user.

ENTRY	Purpose
LDFOPN	to open the KEDAK direct access file by physically opening it and reading the basic tables of the declaration part.
LDFLOC IDFLOC	to locate a given data type by reading and using the associated pointer tables. To provide first data item.

ENTRY	Purpose
LDFNXT IDFNXT	to provide the following data item until last one is found. Requires that first item initially has been located by LDFLOC.
LDFITN	to provide the list of names of data types available on the KEDAK file for the requested material by reading the type address table.
LDFNAM	to provide informations for types with more than two names from the address table of further names.

Table 1: The entries called by the application program

The description of the argument lists of these entries is done in section 3.2, which is entitled "Detailed description of the user called subroutines". The auxiliary subroutines are described in section 3.3.


Declaration part of the KEDAK-file
File identification, date, pointers to ① ② ③
1
2
① material conversion table
② type conversion table
3
③ material address table points to the type address table of the specified material
the type address table for each material points to the data item (no further names)or to further names address table.
with further names
no further names
further names address table
Data part of the KEDAK-file
data items

The data items are constructed by NARG arguments and NFV function values, with NARG = 0,1,2,3,4..., NFV = 0,1,2,3,4... and at least one of the values NARG or NFV must be different from zero.

Fig. 3: The hierarchy of pointers

2. User's guide

NOTE: The user of LDFPAC and IDFPAC is the programmer of any application program,which makes use of the packages to retrieve information from the KEDAK-library.

He is the programmer in sense of section 3 only if he needs to change the subroutines of the package, nevertheless the detailed description could give the user information for effective use of the subroutines in the application program.

2.1 Purpose and use of LDFOPN, LDFLOC and LDFNXT

The LDFOPN subroutine must be called before any other call to a retrieval package subroutine. A call to LDFOPN connects the direct access KEDAK-file and provides the material conversion table, the type conversion table, and the contents and directory for each material on the file for use in the retrieval package (via common/LDFMT/).The LDFOPN routine can be called by the program for a maximum of five different KEDAK-files.

An example forthe use of LDFOPN is shown in the program for printing the contents of a KEDAK-file (2.2.3).

The LDFLOC subroutine is called for each material and data type to locate the first data item. Remember: A data item consists of NARG arguments and NFV functional values. NARG and NFV depend on the specific data type. For example: if a function of the form y = f(x) is stored, both NARG and NFV have the value one. By a LDFLOC- or LDFNXT-call only one data item is retrieved and not all points of a data type. Therefore very few main storage is needed to retain the result of one access to the library. If the data type has further names, LDFLOC is to be called for each name combination, to locate the first data item of this combination.

Each following data item for one data type is provided by a call to the LDFNXT subroutine.

The use of the retrieval subroutines LDFLOC and LDFNXT is shown in the source listing of the subroutine RETXS (2.1.1). RETXS is an application retrieval routine which provides the data from a KEDAK-file within a selectable energy range /EMIN, EMAX/.

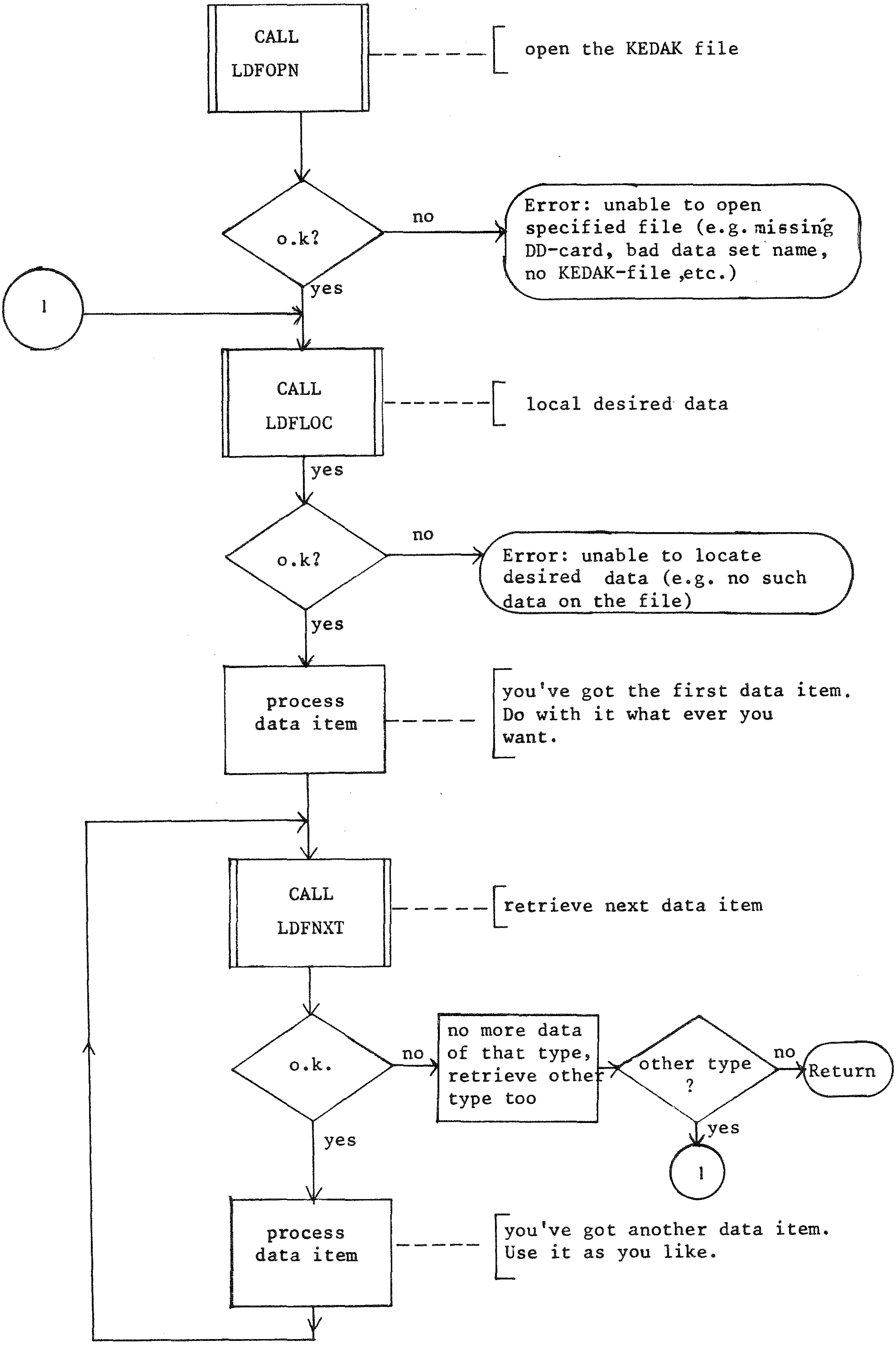

CALL
LDFOPN
open the KEDAK file
o.k?
no
Error: unable to open specified file (e.g. missing DD-card, bad data set name, no KEDAK-file ,etc.)
yes
1
CALL
LDFLOC
local desired data
yes
o.k?
no
Error: unable to locate desired data (e.g. no such data on the file)
yes
process data item
you've got the first data item. Do with it what ever you want.
CALL
LDFNXT
retrieve next data item
o.k.
no
no more data of that type, retrieve other type too
other type ?
no
Return
yes
1
yes
process data item
you've got another data item. Use it as you like.

Fig. 4: Simplified flow chart for the use of LDFLOC, LDFNXT and LDFOPN

Further application retrieval routines are collected in the RETPAC retrieval package (see reference 4 resp. part III.4 of this Compendium).

Note that RETXS does not contain a call to LDFOPN. Therefore a LDFOPN call must be performed in the user's program before RETXS is called.

2.1.1 RETXS - AN EXAMPLE FOR THE USE OF LDFLOC,LDFNXT

```
C
C
      SUBROUTINE RETXS(NARG,NAMES,EMIN,EMAX,X,Y,NUMX,MAXNUM,NR)
C
C     RETXS RETRIEVES KEDAK-DATA: SINGLE VALUED, ONE ARGUMENT,FOR THE
C     ENERGY RANGE (EMIN,EMAX),STARTING WITH THE FIRST DATA ITEM LESS OR
C     EQUAL EMIN, ENDING WITH THE FIRST DATA ITEM GREATER OR EQUAL EMAX.
C     THE USER MAY INTERPOLATE TO THE ENERGY LIMITS, THEREFORE, IF HE
C     NEEDS TO.
C
C     EXOTIC CASES
C     1. NO DATA GREATER OR EQUAL EMIN:
C        THE LAST DATA POINT IS TRANSFERED
C     2. NO DATA LESS OR EQUAL EMAX:
C        FIRST DATA POINT IS TRANSFERED
C     IN CASES 1. OR 2. THE USER MAY EXTRAPOLATE IF HE PREFER TO DO SO.
C
C
C     INTEGER*NARG(3):
C     NARG(1)= THE NUMBER OF NAMES FOR THE REGUESTED DATA TYPE
C     NARG(2)= THE NUMBER OF ARGUMENTS FOR THAT TYPE
C     NARG(3)= THE NUMBER OF FUNCTION VALUES
C     NARG IS FILLED BY THE CALL
C
C     REAL*8 NAMES(N), N=/4,5/
C     NAMES(1)= ISOTOPE NAME (FIRST NAME)
C     NAMES(2)= REACTION TYPE NAME (SECOND NAME)
C     NAMES(3-N)=EVENTUAL FURTHER NAMES
C
C     X  -IS FILLED AFTER EXECUTION OF THE CALL WITH THE ARGUMENTS
C     Y  -IS FILLED AFTER EXECUTION OF THE CALL WITH FUNCTION VALUES
C     NUMX= NUMBER OF DATA POINTS READ INTO X,Y-ARRAYS
C     MAXNUM= MAXIMUM NUMBER OF POINTS THAT MAY BE STORED INTO X,Y
C     NR = RETURNCODE
C
      COMMON /SIGSAV/Z
      DIMENSION X(1),Y(1),Z(2),NARG(1),W(2),NAMES(1),NAMSV(4)
C
C
      ASSIGN 20 TO NST
      I=0
C
C     LOCATE THE DATA TYPE REQUESTED: IDFLOC FOR NUMERIC NAMES
C     PROVIDE THE FIRST DATA ITEM
C     USE LDFLOC FOR ALPHAMERIC NAMES
C
      CALL IDFLOC(NERR,NARG,NAMES,Z)
C
      IF(NERR.EQ.0) GOTO 30
      NAMZ=NARG(1)
      IF(NAMZ.LE.2) GOTO 3
      DO 2 J=3,NAMZ
    2 NAMSV(J-2)=NAMES(J)
C
C     IS THE ARGUMENT FOUND LESS THAN EMIN
   3  IF(Z(1).LE.EMIN) GOTO 5
C
C     IS THE ARGUMENT FOUND GREATER THAN EMAX
```

```
      IF(Z(1).GE.EMAX) GOTO 32
      GOTO 21
C
C     READ THE NEXT DATA ITEM: IDFLOC FOR NUMERIC NAMES
C     USE LDFLOC FOR ALPHAMERIC NAMES
C
    5 CALL IDFNXT(NERR,NARG,NAMES,W)
C
      IF(NERR.EQ.0) GOTO 23
C     IF THE ARGUMENT FOUND IS LESS THAN EMIN, WE IGNORE THIS DATA ITEM.
      IF(W(1).LE.EMIN) GOTO 10
C
C     IS THE ARGUMENT FOUND GREATER THAN EMAX? WE'VE DONE OUR GOAL, IF
C     YES
      IF(W(1).GE.EMAX) GOTO 36
C
C     STORE DATA FOUND FOR REQUESTED ENERGY RANGE
    7 I=I+1
      X(I)=Z(1)
      Y(I)=Z(2)
      I=I+1
      X(I)=W(1)
      Y(I)=W(2)
      GOTO NST,(20,200)
C
C     READ THE NEXT DATA ITEM: IDFLOC FOR NUMERIC NAMES
C     USE LDFLOC FOR ALPHAMERIC NAMES
C
   10 CALL IDFNXT(NERR,NARG,NAMES,Z)
      IF(NERR.EQ.0) GOTO 26
C
C     IS THE ARGUMENT FOUND LESS THAN EMIN
      IF(Z(1).LE.EMIN) GOTO 5
C
C     IS THE ARGUMENT FOUND GREATER THAN EMAX
      IF(Z(1).GE.EMAX) GOTO38
C
C     STORE DATA FOUND FOR REQUESTED ENERGY RANGE
   11 I=I+1
      X(I)=W(1)
      Y(I)=W(2)
   12 I=I+1
      X(I)=Z(1)
      Y(I)=Z(2)
      GOTO NST,(20,200)
C
C     READ THE NEXT DATA ITEM: IDFLOC FOR NUMERIC NAMES
C     USE LDFLOC FOR ALPHAMERIC NAMES
C
   20 CALL IDFNXT(NERR,NARG,NAMES,Z)
      IF(NERR.EQ.0) GOTO 22
C
C     IS THE ARGUMENT FOUND GREATER THAN EMAX
      IF(Z(1).GE.EMAX) GOTO 24
      IF(I.EQ.MAXNUM) GOTO 34
C
C     STORE DATA FOUND FOR REQUESTED ENERGY RANGE
   21 I=I+1
      X(I)=Z(1)
      Y(I)=Z(2)
      GOTO 20
C
C
C     CONTINUE READING: IF THE NUMBER OF DATA POINTS IN (EMIN,EMAX)
C     IS GREATER MAXNUM, RETURNCODE=2 AFTER RETXS-CALL.
C     USE THIS ENTRY TO CONTINUE RETRIEVAL OF THE SUBSEQUENT DATA ITEMS.
C
      ENTRY REPXS(NARG,NAMES,EMIN,EMAX,X,Y,NUMX,MAXNUM,NR)
C
      I=0
      NAMZ=NARG(1)
      IF(NAMZ.LE.2) GOTO 21
      DO 19 J=3,NAMZ
   19 NAMSV(J-2)=NAMES(J)
      GOTO 21
C
   22 IF(I.LT.1) GOTO 23
C
C     THE LAST DATA POINT FOR THE CURRENT TYPE HAS BEEN STORED IN X,Y
      NR=1
      GOTO 198
C
C     THE ARGUMENT OF LAST DATA POINT IS STILL LESS THAN EMIN. THIS DATA
C     POINT IS TRANSMITTED
   23 NR=5                                                              R
      I=1
      X(I)=Z(1)
      Y(I)=Z(2)
      GOTO 198
C
C     ARGUMENT OF THE LAST  DATA POINT ALREADY GREATER EMAX. THIS DATA
C     POINT IS TRANSMITTED
   24 IF(I.EQ.MAXNUM) GOTO 34
      I=I+1
      X(I)=Z(1)
      Y(I)=Z(2)
C
C     THE LAST DATA POINT FOR THE CURRENT TYPE HAS BEEN STORED IN X,Y
      NR=0
      GOTO 200
C
C
C     THE ARGUMENT OF LAST DATA POINT IS STILL LESS THAN EMIN. THIS DATA
C     POINT IS TRANSMITTED
   26 I=1
      X(I)=W(1)
      Y(I)=W(2)
      NR=5
      GOTO 198
C
C
C     NO DATA FOR REQUESTED TYPE
   30 NR=3
      GOTO 200
C
C
C
C     ARGUMENT OF THE FIRST DATA POINT ALREADY GREATER EMAX. THIS DATA
```

```
C      POINT IS TRANSMITTED
   32 NR=4
      I=1
      X(1)=Z(1)
      Y(1)=Z(2)
      GOTO 200
C
C      MAXNUM OF DATA POINTS HAVE BEEN FILLED INTO X,Y WITHOUT READING
C      THE END OF THE DATA TYPE. CONTINUE THE READING BY A CALL TO REPXS
   34 NR=2
      GOTO 200
C
C
C      THE LAST DATAPOINT FOR THE REQUESTED RANGE HAS BEEN READ
   36 ASSIGN 200 TO NST
      NR=0
      GOTO 7
C
   38 ASSIGN 200 TO NST
      NR=0
      GOTO 11
C
C
  198 IF(NAMZ.LE.2) GOTO 200
      DO 199 J=3,NAMZ
  199 NAMES(J)=NAMSV(J-2)
C
  200 NUMX=I
      RETURN
      END
```

2.2 Purpose and use of LDFITN, LDFNAM

The subroutines LDFITN and LDFNAM are called by the user for special purposes: LDFITN provides the list of different data types available on the KEDAK-file for the requested material. If a data type has further names LDFNAM may be called to retrieve the further names for that type (e.g. a list of level energies for inelastic excitation). Thus LDFITN and LDFNAM are very useful, when information on the contents of KEDAK is desired.

The use of LDFITN is shown in the subroutine FILLTP (2.2.1). FILLTP selects from data types available on the KEDAK-file only those types comprised in the list given by the array TYPS and stores them in the arrays TYP, ITYPE for use by the calling program.

The program for printing the contents of a KEDAK-file shows the use of LDFNAM, LDFITN and LDFOPN (2.2.3).

2.2.1 FILLTP - AN EXAMPLE FOR THE USE OF LDFITN

```
C
C
      SUBROUTINE FILLTP(NR,MAT,TYP,ITYP,NT)
C
C     FILL TABLE OF SINGLE VALUED, ENERGY DEPENDENT TYPES AVAILABLE FOR
C     THE ISOTOPE MAT ON THE KEDAK-FILE
C
C     NR -RETURNCODE
C     NR =1    NO ERROR
C     NR =0    ERROR
C     MAT -MATERIAL NAME                                        :FIRST NAME
C     TYP -ARRAY TO STORE THE ALPHAMERIC REACTION TYPE NAMES :SECOND NAM
C     ITYP-ARRAY TO STORE THE NUMERIC REACTION TYPE NAMES
C     NT  -NUMBER OF NAMES FILLED INTO TYPE,ITYPE
C
C
C     TYPS-COMPLETE LIST OF DATA TYPES THAT ARE SINGLE VALUED. (CROSS
C     SECTIONS, CROSS SECTION RATIOS, ETC.)
C
      REAL*8 TYPS(32)/'SGT','SGN','SGX','SGI','SGIZC','SGIZ','SG2N','SG3
     1N','SGIA','SGI3A','SG2NA','SG3NA','SGIP','SGNI','SGA','SGF','SGG',
     2'SGP','SGD','SGH3','SGHE3','SGALP','SG2HE','SGTR','MUEL','ETA','AL
     3PHA','NUE','NUEP','CHIF','CHIFD','SGNC'/,TYP(70),MAT,NAMES(3)
      INTEGER NP(70),MTYP/70/,MTYPS/32/,NARG(3)
      DIMENSION ITYP(70)
      COMMON/LDFTT/MO(3)
C
      KOUT=6
C
C     FETCH THE TABLE OF REACTION TYPES AVAILABLE FOR THE REQUESTED
C     ISOTOPE ON THE KEDAK-FILE - ALPHAMERIC NAMES
C
      CALL LDFITN(NR,MAT,TYP,NP,NT,MTYP,2)
C
C     TYP   - TABLE OF SINGLE VALUED REACTION TYPES
C     NP    -NUMBER OF DATA POINTS FOR THE REACTION TYPE
C     NT    -NUMBER OF AVALABLE TYPES
C     MTYP -MAXIMUM NUMBER OF TYPES - DIMENSION FOR TYP,ITYPE ARRAYS
C     2     -ALPHAMERIC NAMES
C
      IF(NR.EQ.0.AND.NT.EQ.0) GOTO 100
      IMAT=MO(3)
C
C     FETCH THE TABLE OF REACTION TYPES AVAILABLE FOR THE REQUESTED
C     ISOTOPE ON THE KEDAK-FILE - NUMERIC NAMES
C
      CALL LDFITN(NR,IMAT,ITYP,NP,NT,MTYP,1)
C
C     IMAT -NUMERIC MATERIAL NAMES
C     1     -NUMERIC NAMES
C
      IF(NR.EQ.0.AND.NT.EQ.0) GOTO 99
      J=0
      DO 12 N=1,NT
      DO 10 K=1,MTYPS
C
C     IS THE TYPE FOUND SINGLE VALUED ?
C
```

```
      IF(TYP(N).NE.TYPS(K)) GOTO 10
      J=J+1
      TYP(J)=TYP(N)
      ITYP(J)=ITYP(N)
      NP(J)=NP(N)
      GOTO 12
   10 CONTINUE
C
   12 CONTINUE
      NT=J
      GOTO 100
   99 WRITE(KOUT,600)
  600 FORMAT(///'   PROGRAMMING-ERROR IN LDFITN.CALL PROGRAMMER.')
      STOP
  100 CONTINUE
      RETURN
      END
```

2.2.2 Job control statements for the printing program

```
//INR048KE JOB (0048,101,P6M1A),LANGNER,CLASS=A,REGION=300K
/*SETUP DEVICE=2314,ID=GFK016
/*SETUP DEVICE=2314,ID=GFK050
// EXEC FHCLG,PARM.C='MAP',PARM.L='MAP,LIST'
//C.SYSIN DD *
/*
//L.LIBA DD UNIT=2314,VOL=SER=GFK029,DSN=INR.STEIN.LOAD,DISP=SHR
//L.SYSIN DD *
 INCLUDE LIBA(IDFPAC)
//G.FT01F001 DD UNIT=2314,VOL=SER=GFK050,DSN=KEDAK3,DISP=SHR
//G.SYSIN DD DUMMY
/*
//
```

2.2.3 A PROGRAM TO PRINT THE CONTENTS OF A KEDAK-FILE.

AN EXAMPLE FOR USE OF LDFOPN, LDFITN, AND LDFNAM.

```
C
      COMMON /LDFTC/ IDTYC(4),TYPNAM(70),ITYP(70),LTYCON
      COMMON /LDFMT/ IDMAT(4),MATNAM(120),IMAT,NTYP(120),ATYP(2,120),
     *LMATAB
      COMMON /LDFTT/ MO(3)
      COMMON /LDFIL/ IFIL,LREC,LR
      COMMON Z,ZZ(302),NOUT
C
      REAL*8 MATNAM,TYPNAM,MANAM(70),TYPES(70),NAMES(5),Z(300)
C
      DIMENSION IMANAM(120),IDX(3),AMACON(2),ATYCON(2),AMATAB(2),ITYPES(
     *70),XNAM(200),NP(70),IMA(1),IMAT(120),JC(20)
      DATA JC/7,9,12,17,22,27,35,38,43,45,47,49,51,52,54,55,56,57,59,62/
      DATA NT/70/
      INTEGER*2 NTYP,ATYP
C
C
      IFILE=1
      NOUT= 6
      NPRINT=0
      NAMES(3)=0.
      NAMES(4)=0.
      NAMES(5)=0.
      MAX=200
C
C
C
C     OPEN THE KEDAK-FILE WITH THE DATASET REFERENCE NUMBER IFILE
C     LOAD INFORMATION FROM THE MATERIAL CONVERSION TABLE, THE TYPE CON-
C     VERSION TABLE AND THE MATERIAL ADDRESSTABLE INTO THE LDFTC COMMON
C     AND LDFMT COMMON
C
      CALL LDFOPN(IFILE,NDATE,&1000)
C
      LR1=LR
      READ (IFILE'1) IDX,IDAT,LMACON,AMACON,LTYCON,ATYCON,LMATAB,AMATAB
      LR=LR1
C
C     FETCH THE MATERIAL CONVERSION TABLE
C
      CALL LDFCTB(AMACON,LMACON,MANAM,IMANAM)
C
      WRITE (NOUT,99)
   99 FORMAT(1H1)
      WRITE (NOUT,100) IDX,IDAT
  100 FORMAT(17X,'THE TABLE OF CONTENTS OF THE KEDAK FILE',///10X,'FILE
     1IDENTIFICATION :',3A4,4X,'DATE :',I9,//////)
C
C
C     PRINT MATERIAL CONVERSION TABLE
C
      WRITE (NOUT,101) LMACON
  101 FORMAT (8X,'THE MATERIAL CONVERSION TABLE CONTAINS',I4,' MATERIAL
     1NAMES',/////)
C
```

```
      WRITE (NOUT,120)
  102 FORMAT(3(1X,' ALPHAMERIC',2X,'NUMERIC'),/)
  120 FORMAT(3(1X,'ALPHAMERIC',1X,'NUMERIC '),/)
      WRITE (NOUT,103) (MANAM(I),IMANAM(I),I=1,LMACON)
  103 FORMAT(3(1X,A8,2X,I7,2X))
C
C
C     PRINT TYPE CONVERSION TABLE
C
      WRITE (NOUT,98)
   98 FORMAT(/////)
      WRITE (NOUT,104) LTYCON
  104 FORMAT('1'////8X,'THE TYPE CONVERSION TABLE CONTAINS ',I4,' DATA T
     *YPE NAMES',/////)
      WRITE (NOUT,102)
      WRITE (NOUT,103)(TYPNAM(I),ITYP(I),I=1,LTYCON)
C
C
C     PRINT THE LIST OF ISOTOPES AVAILABLE ON THE KEDAK-FILE
C
      WRITE (NOUT,98)
      WRITE (NOUT,105) LMATAB
  105 FORMAT(/13X,'ON THE KEDAK-FILE ',I4,' ISOTOPES ARE AVAILABLE :'/
     *////)
      WRITE (NOUT,106) (MATNAM(I),I=1,LMATAB)
  106 FORMAT(1X,7A10)
      WRITE (NOUT,99)
C
C
C     PRINT THE LIST OF DATA TYPES FOR EACH ISOTOPE
C
      JZ=1
      DO 10 J=1,LMATAB
      IF(J.LT.JC(JZ)) GO TO 21
      JZ=JZ+1
      WRITE (NOUT,99)
C
C     FETCH DATA TYPES AVAILABLE FOR THE ISOTOPE - ALPHAMERIC NAMES
   21 CALL LDFITN (NR,MATNAM(J),TYPES,NP,MTYP,NT,2)
C
C     FETCH DATA TYPES AVAILABLE FOR THE ISOTOPE - NUMERIC NAMES
      IMA(1)=MO(3)
      CALL LDFITN (NR,IMA(1)    ,ITYPES,NP,MTYP,NT,1)
C
      WRITE (NOUT,107) (MATNAM(J),MTYP,(TYPES(I),I=1,MTYP))
  107 FORMAT (///1X,'FOR ',A8,I6,' DATA TYPES ARE AVAILABLE : '/  (7(2X,
     *A8)))
C
C
C     PRINT DATA TYPES WITH FURTHER NAMES
C
      IZ=0
      DO 20 K=1,MTYP
      IF(ITYPES(K).EQ.30050) GO TO 11
      IF(ITYPES(K).GE.40021) GO TO 11
      GO TO 20
C
C
   11 IZ=IZ+1
      NAMES(1)=MATNAM(J)
      NAMES(2)=TYPES(K)
C
C     FETCH THE FURTHER NAMES OF THF DATA TYPE
C
      CALL LDFNAM (NR,NAMES,XNAM,N1,N2,MAX,2)
C
C
      WRITE (NOUT,108)  TYPES(K),N1,N2
  108 FORMAT(1X,'THE DATA TYPE ',A8,' HAS ',I1,' FURTHER NAMES AND ',I4,
     *' NAME COMBINATIONS : ')
      NN=N1*N2
      WRITE (NOJT,109) (XNAM(I),I=1,NN)
  109 FORMAT(1X,5E13.6)
C
   20 CONTINUE
   12 LTY=MTYP-IZ
      IF(NPRINT.EQ.0) GO TO 10
      WRITE (NOUT,110)
  110 FORMAT(1X,(6(10X,'NUMBER OF    '))/1X,6( 10X,'DATA POINTS '))
      WRITE (NOUT,111) (TYPES(I),NP(I),I=1,LTY)
  111 FORMAT (6(2X,A8,I10,2X))
C
   10 CONTINUE
 1000 STOP
      END
```

2.3 Storage requirements and availability (internal use)

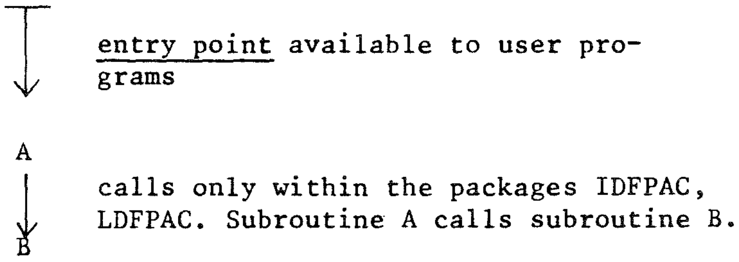
Without overlay (see 2.5) the program size is 24K bytes storage on IBM/370-168 when using the H-extended compiler. For each KEDAK-file accessed additional 8K buffer area is needed. A DD-card must be available for each file.

The source programs of the retrieval packages are stored on magnetic tapes. They can be made available by the authors or the Karlsruhe Nuclear Data Evaluation Group. Object modules are available as the members IDFPAC or LDFPAC in the partitioned dataset INR.STEIN.LOAD on the disk GFK029. An example for linking the IDFPAC object module to an application program is given in (2.2.2) with the job control statements for the contents printing program (see 2.2.3).

Foreign user's should contact the authors or the Karlsruhe Nuclear Data Evaluation Group.

2.4 The call scheme in IDFPAC, LDFPAC

Explanation:



entry point available to user programs

calls only within the packages IDFPAC, LDFPAC. Subroutine A calls subroutine B.

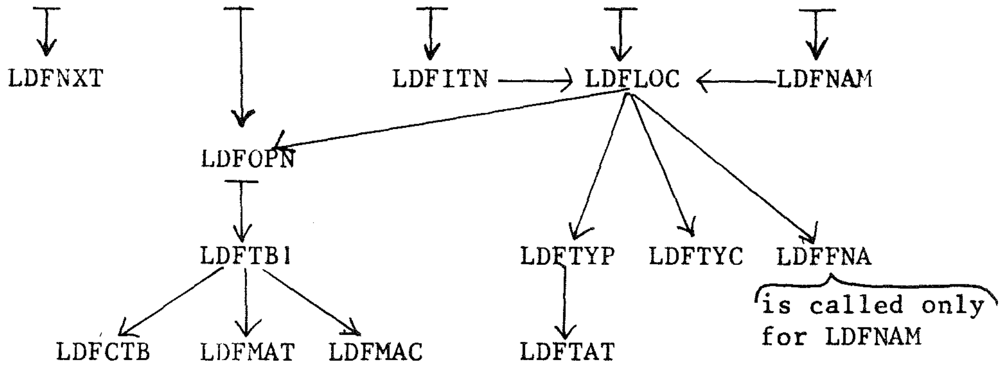


Fig. 4: CALL scheme

The subroutines LDFCTB, LDFMAT, LDFMAC are only used if the information in the associated common blocks has to be filled for the first time or if this information has been destroyed by overlaying or if data are retrieved from more than one KEDAK-file and the information in the common blocks has to be switched to the file now in use. Almost all subroutines call the LDFREC subroutine. The call scheme shows how and when the auxiliary subroutine are involved by the user called subroutines or internally. The sequence of calls for the user called subroutines is determined by the problem solved in the users program. The only restrictions are: LDFOPN must be called before any other call to a retrieval routine for that KEDAK-file. Before the first LDFNXT call a LDFLOC call has to be performed for the requested data type.

Possible call sequences for user called retrieval routines:

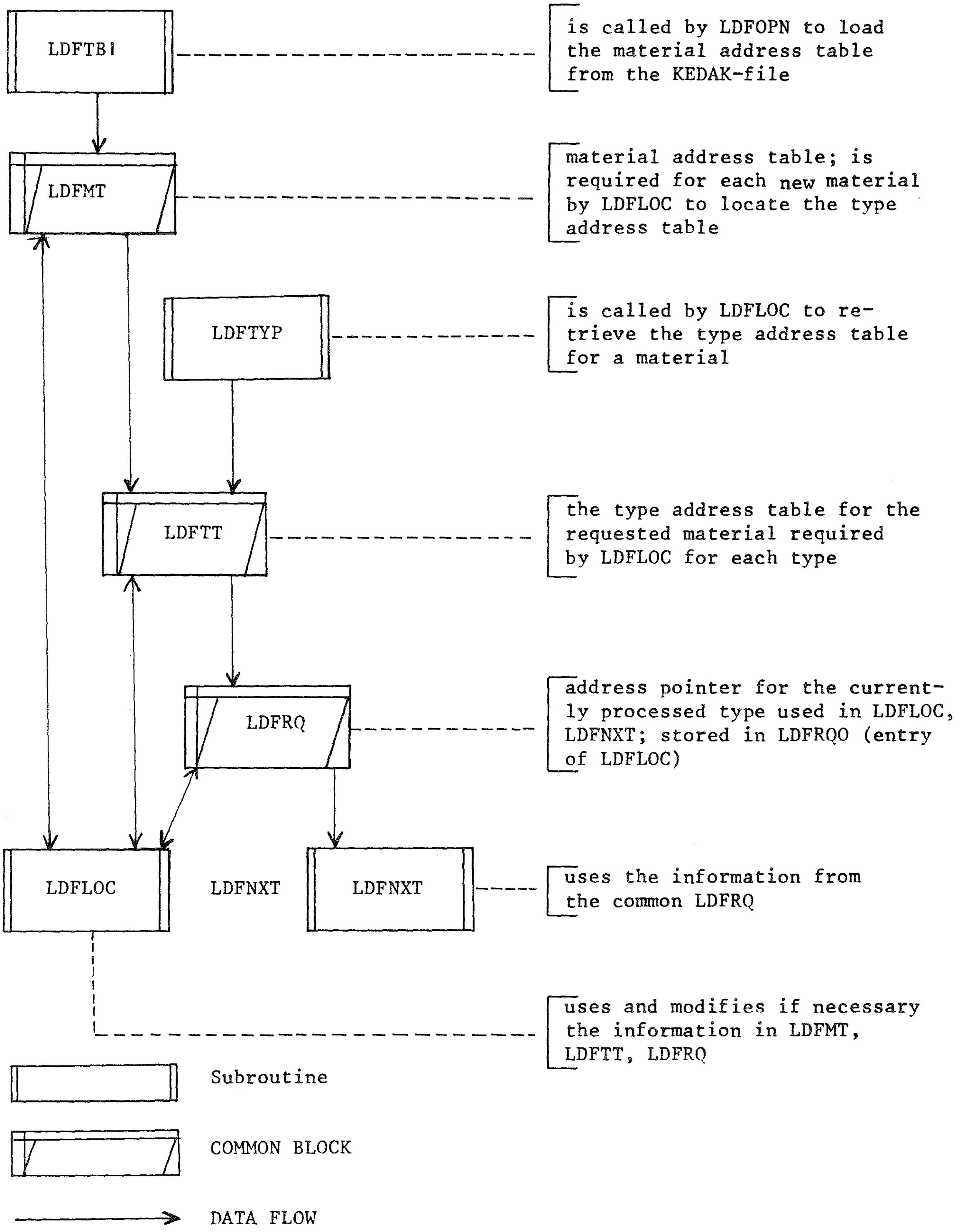
1 LDFOPN, LDFLOC, LDFNXT, LDFNXT, LDFNXT,...

2 LDFOPN, LDFLOC, LDFNXT, LDFNXT,...,LDFNXT, LDFLOC, LDFNXT, LDFNXT ...

3 LDFOPN, LDFITN, LDFNAM, LDFLOC, → 1

The retrieval packages IDFPAC and LDFPAC retrieve for internal use the addressing information and the contents of the KEDAK-library and retains this information in labeled common blocks.

The common /LDFMT/ retains the material address table of the processed KEDAK-file. The common /LDFTT/ retains the type address table of the requested isotope (material). The common /LDFRQ/ retains the pointers for the requested data type.



is called by LDFOPN to load the material address table from the KEDAK-file

material address table; is required for each new material by LDFLOC to locate the type address table

is called by LDFLOC to retrieve the type address table for a material

the type address table for the requested material required by LDFLOC for each type

address pointer for the currently processed type used in LDFLOC, LDFNXT; stored in LDFRQO (entry of LDFLOC)

uses the information from the common LDFRQ

uses and modifies if necessary the information in LDFMT, LDFTT, LDFRQ

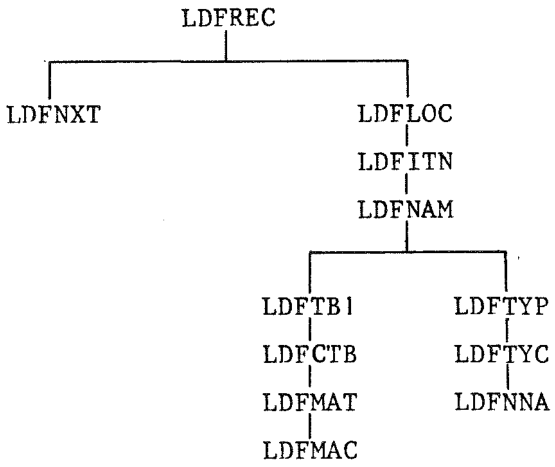
Fig. 5: Hierarchy of common blocks

2.5 Possible overlay structures for IDFPAC and LDFPAC

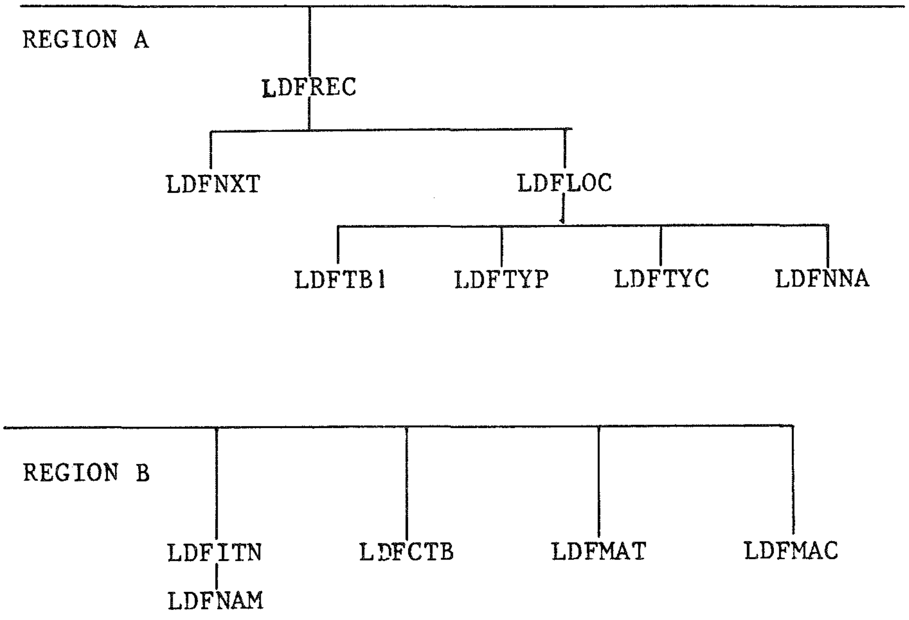
The overlay structure for IDFPAC could consist of one or two overlay regions (see reference 5).

CAUTION: LDFOPN must not be overlayed and must be called before any other retrieval routine (see 3.2.1).

The one region overlay structure - call optimization



The two region overlay structure - common optimization

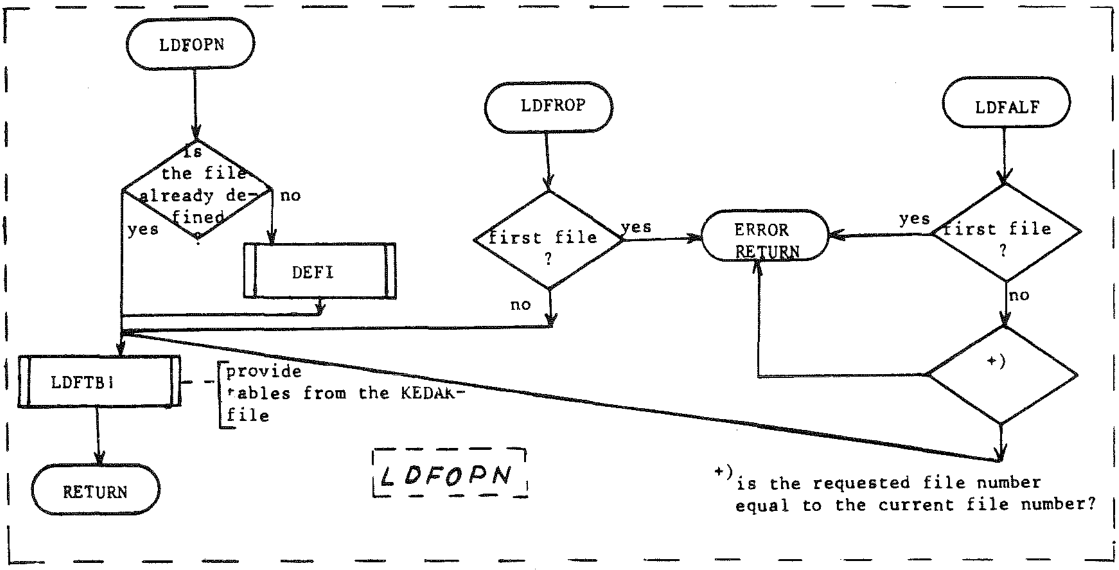


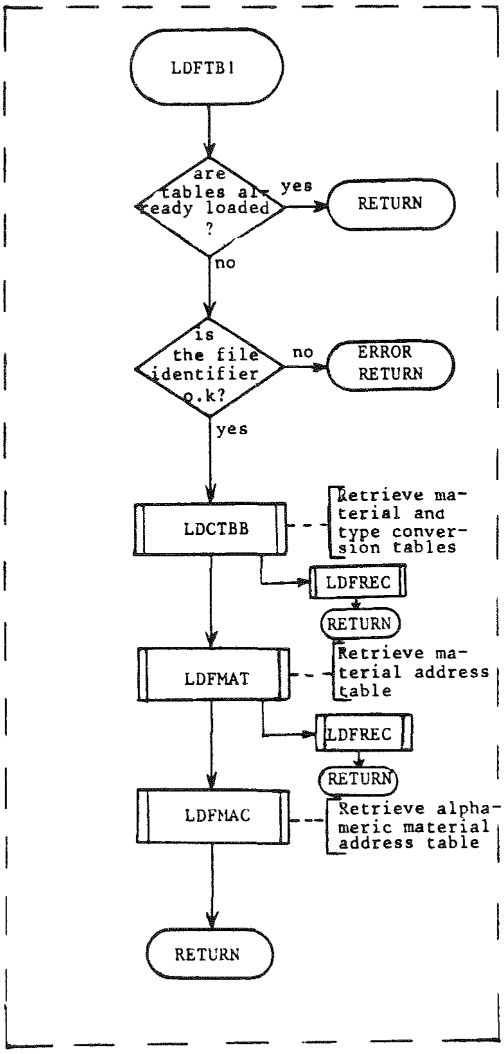
2.6 The unresolved reference DEFI

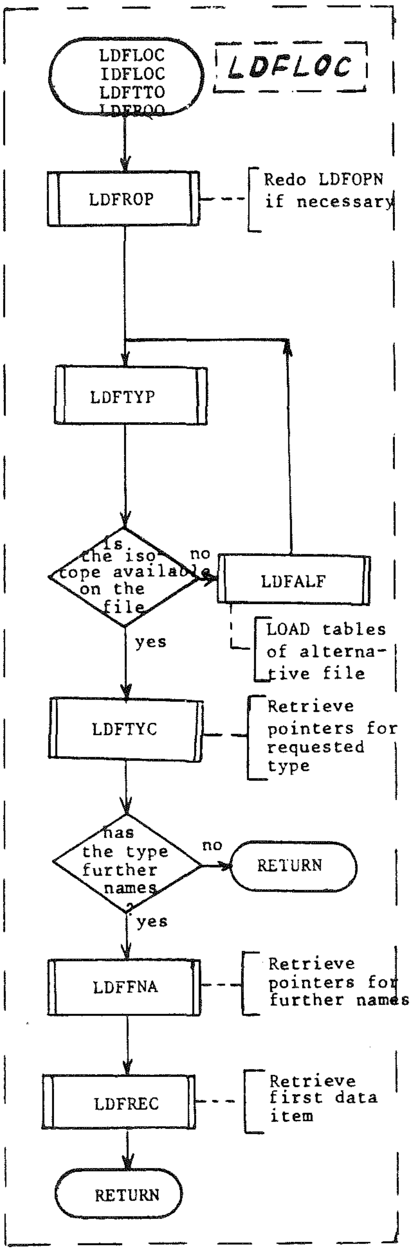
The FORTRAN IV DEFINE FILE statement can only be applied with a fixed file number. In order to avoid this disadvantage and to make the LDFOPN call dynamical with respect to the KEDAK-file number an Assembler subroutine DEFI available at KARLSRUHE is used. DEFI must be supplied separately (see reference 3).

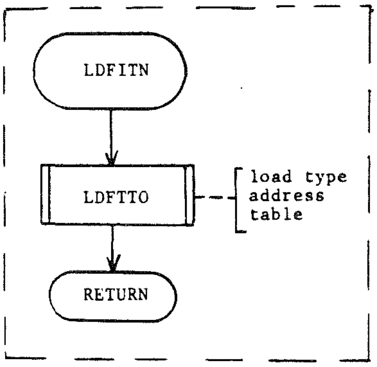
For foreign users a replacement routine can be made available.

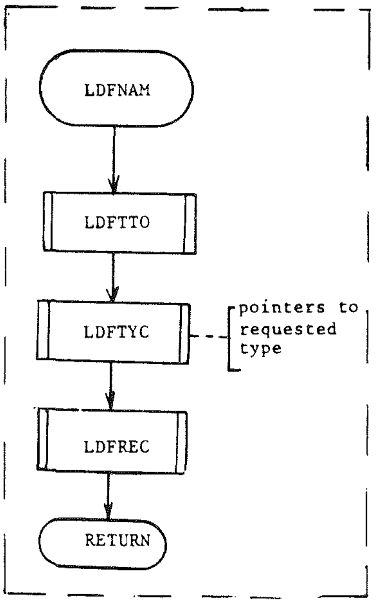
2.7 Flow chart


LDFOPN
is the file already defined ?
yes
no
DEFI
LDFTB1
provide tables from the KEDAK-file
RETURN
LDFOPN
LDFROP
first file ?
yes
no
ERROR RETURN
LDFALF
yes
first file ?
no
+)
+) is the requested file number equal to the current file number?


LDFTB1
are tables already loaded ?
yes
RETURN
no
is the file identifier o.k?
no
ERROR RETURN
yes
LDCTBB
Retrieve material and type conversion tables
LDFREC
RETURN
LDFMAT
Retrieve material address table
LDFREC
RETURN
LDFMAC
Retrieve alphameric material address table
RETURN


LDFLOC
IDFLOC
LDFTTO
LDFROO
LDFLOC
LDFROP
Redo LDFOPN if necessary
LDFTYP
is the isotope available on the file ?
no
LDFALF
LOAD tables of alternative file
yes
LDFTYC
Retrieve pointers for requested type
has the type further names ?
no
RETURN
yes
LDFFNA
Retrieve pointers for further names
LDFREC
Retrieve first data item
RETURN





3. Detailed description

3.1 The labeled common blocks used by IDFPAC and LDFPAC

3.1.1 The common LDFMT

The common block LDFMT provides information from the material address table. The arrays are filled by the subroutine LDFTB1. The LDFMT common is used by the LDFTYP routine (3.3.6) to retrieve the type address table for the requested material.

COMMON /LDFMT/ IDMAT, MATNAM, IMAT, NTYP, LMATAB

IDMAT(1)
.
.
.
IDMAT(4) — is a REAL*4 array that contains the common block label as an alphatext 'KEDABIBLIOTHnnn' nnnn varies from '1111' to '5555' and is an indicator internally used to relate the common block label with the KEDAK-file number whose tables it contains.

MATNAM(120) — is a REAL*8 array to retain the alphameric names of the materials available on the KEDAK-file just being processed.

IMAT(120) — is an INTEGER*4 array to retain the numeric names of the materials available on the KEDAK-file just being processed.

NTYP(120) — is an INTEGER*2 array to retain the number of reaction types for each material.

ATYP(2,120) — is an INTEGER*2 array to retain the pointers to the type address table for each material (starting address of each material).

LMATAB — is an integer variable that specifies the number of materials available on the KEDAK-file just processed.

3.1.2 The common LDFTT

The common LDFTT provides the type address table for one material. The LDFTAT subroutine (3.3.5) loads the information from the declaration part of the KEDAK-file into the common.

```
COMMON /LDFTT/ MAT(3), ITYNAM(70), ITYTAB(6,70) LTYTAB
```

MAT(1)
MAT(2) — are variables to retain the alphameric name of the material requested currently.

MAT(3) — is a variable to retain the numeric name of the material.

ITYNAM — is an integer array to retain the numeric names of the data types available on the KEDAK-file for the material MAT

ITYTAB — is an INTEGER*2 array to retain the address pointers and related information for each type. ITYNAM and ITYTAB are filled by LDFTAT.

ITYTAB(1,I) — specifies the number of further names for the type ITYNAM(I).

ITYTAB(2,I) — specifies the number of arguments.

ITYTAB(3,I) — specifies the number of functional values.

ITYTAB(4,I) — specifies the number of data points.

ITYTAB(5,I) — specifies the record number.

ITYTAB(6,I) specifies the word number within the record ITYTAB(5,I) and ITYTAB(6,I) are called an address (see 1.3.2).

LTYTAB is an integer variable that indicates the number of data types available on the KEDAK-file for the material MAT.

The following subroutines are using the LDFTT common:

LDFNAM, LDFTYC, LDFLOC, LDFTYP, LDFITN, LDFTAT.

3.1.3 The common LDFRQ

The LDFRQ common retains the names and addressing information of the currently requested data type, that is the pointers to the data array.

```
COMMON /LDFRQ/ MAT(3), TYP(3), FNAM(4), NAM(6), NFNAM(3), NRDONE, NFDØNE
```

MAT(1)
MAT(2) } are real variables to retain the alphameric name of the material currently requested.

MAT(3) retains the numeric name of the material.

TYP(1)
TYP(2) } are real variables to retain the alphameric type name currently requested.

TYP(3) retains the numeric type name.

NOTE: The names are stored by the LDFNAM routine (3.2.5). They can be changed by LDFLOC (3.2.2) and LDFTYC (3.3.9).MAT and TYP are called the first two names.

FNAM(4) is an array that retains the further names if a data type specified has more than two names. It is filled by the LDFNAM routine and can be changed by LDFNXT (3.2.3).

NAM is an integer array retaining information from the type address table.

NAM(1) indicates the number of further names in the specified name combination MAT, TYP. It is stored by the LDFTYC routine and can be changed by LDFNXT.

NAM(2) specifies the number of arguments for the data item requested, or by types with further names: the number of further names. It is stored by LDFFNA.

NAM(3) specifies the number of function values for the current type.

NAM(4) specifies the number of data items for the requested name combination or the number of entries in the further name table if any. It is stored by LDFDPN.

NAMES(5)
NAMES(6) } specifies the starting address for the data of this type or the address of the further name table if there are any (record/word).

In case of more than two names (further names)

NFNAM(1) specifies the number of data items of the currently requested name combination.

NFNAM(2)
NFNAM(3) } specify the starting address (record/word) of the data items for the current name combination.

ATTENTION: for consistency a call to LDFLOC will exchange NAM(4-6) with NFNAM(1-3) and modify their meaning correspondingly!!!

NRDONE is an integer variable that indicates the number of data items already processed.

NFDONE is an integer variable that indicates the number of further name combinations already processed.

The LDFRQ common is used in the subroutines LDFLOC, LDFNAM, LDFFNA, LDFTYC, LDFNXT.

3.1.4 The common LDFRE

The LDFRE common retains the last record read from the KEDAK-file. The LDFRE common is used by the following subroutines: LDFLOC, LDFNXT, LDFNAM, LDFCTB, LDFMAT, LDFFNA, LDFTAT, LDFTYP.

COMMON /LDFRE/ ID, W(880), LW

ID — is a REAL*8 variable to store the alphatext identification: 'KEDABIBL'.

W — is a REAL*4 array to store the record read from the file.

LW — is an integer variable $\leq$ LRECD (see common LDFIL (3.1.5)) that points to the word in the field W to be processed next. LW is set to one by LDFREC (see 3.3.8).

3.1.5 The common LDFIL

The LDFIL common contains information about the KEDAK-file just processed.

COMMON /LDFIL/ IFIL, LRECD, LR

IFIL — is an integer variable that specifies the data set reference number for the KEDAK-file currently open (= whose tables reside in the respective common).

LRECD — is an integer variable that specifies the maximum size of each record on IFIL.

LR — is an integer variable specifying the record on IFIL to be transmitted by the next READ-statement (associated variable).

LDFIL is used by the subroutines: LDFOPN, LDFLOC, LDFNXT, LDFNAM, LDFFNA, LDFCTB, LDFMAT, LDFTYP, LDFERR, LDFREC, LDFTB1.

3.1.6 The common LDFTC

The LDFTC common retains the type conversion table from the KEDAK-file specified by IDTYC. It is filled by the LDFTB1 subroutine (3.3.1).

COMMON /LDFTC/ IDTYC(4), TYPNAM(70), ITYP(70), LTYCON

IDTYC see common LDFMT (3.1.1) description for IDMAT.

TYPNAM is a REAL*8 array retaining the alphameric type names.

ITYP is an integer array retaining the corresponding numeric type names.

The common LDFTYC is used by the subroutines LDFITN, LDFTYC.

3.1.7 The common LDFRC

The LDFRC common provides information for the LDFERR error diagnostic routine (3.2.6).

COMMON /LDFRC/ NRETCD, NXF, MAT(3), TYP(3), XF(4)

NRETCD is an integer variable that indicates the error message.

NXF is an integer variable that specifies the number of further names.

MAT(1)
MAT(2) } are variables to retain the alphameric material name

MAT(3) and the numeric material name.

TYP(1)
TYP(2) } are variables to retain the alphameric type name

TYP(3) and the numeric type name.

XF is an array to retain further names of the material and data type for which the message is to be printed.

The LDFRC common is used in the subroutines LDFERR, LDFOPN, LDFLOC, LDFNXT, LDFMAT, LDFTAT, LDFTYC, LDFTYP, LDFFNA, LDFTB1.

3.2 Detailed description of the user called subroutines LDFOPN, LDFLOC, LDFNXT, LDFITN, LDFNAM

3.2.1 Subroutine LDFOPN

The subroutine LDFOPN defines the KEDAK-file and provides the first three tables from the declaration part of the KEDAK dataset:

1. conversion table for material names,
2. conversion table for reaction type names,
3. contents and directory for each material.

The file numbers of the opened files together with some other important information (e.g. maximum number of different KEDAK files that can be used parallel) is kept in LDFOPN.

Therefore CAUTION: The LDFOPN subroutine must not be overlayed. The LDFOPN subroutine must be the first subroutine called by a program.

The direct access files are defined by using the DEFI subroutine (see 2.6).

Although a maximum of five different KEDAK-datasets can be simultaneously opened by the retrieval packages the LDFOPN subroutine may be repeatedly called for each file by the program.

The call:
CALL LDFOPN (IFIL, IDAT, εLABEL)

IFIL is an integer variable or constant representing the dataset reference number (xx) that was specified in the name field of the data definition /DD/ job control statement for the KEDAK-dataset on the device FTxxF001. The value may be any integer number from 1 to 50 (for Karlsruhe).

IDAT is an integer variable in which the subroutine returns date of the last change of the KEDAK-file.

& LABEL statement number in the calling program, where the processing has to continue after return from LDFOPN, in the case of an error in LDFOPN.

The LDFOPN subroutine includes two additional entries restricted only to internal package use:

Entry LDFROP (IFI, *)
called by: LDFLOC (3.2.2)
to reread the tables of the current file

IFI returns the file reference number of the current file

* label (see above)

LDFROP is entered when the package detects that information in the common blocks filled by the open process has been destroyed (e.g. overlayed).

Entry LDFALF (IFS, IFI, *)
called by: LDFLOC
to load the tables of an alternative file (if there is any).

LDALF is called when the package searching for a material specified by LDFLOC, is unable to locate it on the KEDAK-file currently in use. LDFALF subsequently will switch to the next logical file if there is any, thus providing an automatic search mechanism within all opened KEDAK-files. Searching stops if wrap-around leads back to the file where the search has started. The logical sequence is the same as those defined by the sequence of the user LDFOPN calls. The user may avoid searching by executing an LDFOPN call to the file which contains the desired material before calling LDFLOC(always recommended in cases where several files are in simultaneously use).

The arguments

IFS, IFI — are integer variables or constants that specify dataset reference numbers for KEDAK-files as IFIL in the LDFOPN argument list.

IFS — gives the file reference number of the file on which the search has started. The calling program supplies its value and is responsible for not modifying it during the search process.

IFI — returns the file reference number of the file of which the tables have been loeaded and where the search is continued.

Each time the routine is called the pointer to the internally maintained table of the open file reference numbers is incremented by one and the associated tables are loaded. If the last entry within the table is reached, wrap around occurs (pointer is set to entry number one). The file number pointed to is loaded into IFI.

If IFI matches IFS no loading of tables is performed and the error return is used. IFI will take the value of the file reference number of the KEDAK-file currently open.

Example:

1. only one file open
   file reference number 1
   Enter with IFS = 1.
   Since no other files are open, IFI will receive
   the value 1 and matches IFS. No further action
   takes place and error return is used.

2. Two files open: file reference number 1 and
   file reference number 5, current file is that with
   number 5.
   Enter with file number 5. Search open file table.
   file number 1 is found, IFI = 1, load tables for
   file number 1, take normal return.
   Assume LDFLOC does not find requested material

on this file either. It calls LDFALF again with IFS = 5 still. Search open file table. Next file number is 5 = > IFI. Search started with IFS = 5 which matches IFI. IFI is readjusted to file number 1 (currently open), perform no further action (especially no table loading) and error return is taken.

Externals used:

The LDFOPN subroutine uses the labeled common blocks /LDFIL/ and /LDFRE/ (see 3.1.5/4). A call to the LDFTB1 subroutine (see 3.3.1) fills into the common /LDFMT/ (see 3.1.1) alphameric and numeric material names and the number of materials on the KEDAK-file. The program size is 1008 bytes for the H-extended compiler on IBM 37o/168

3.2.2 Subroutine LDFLOC

The subroutine LDFLOC locates the data of requested material and data type and provides the first data item of this type. Whenever a new reaction type is to be retrieved or one of the names is changed,LDFLOC must be called.

The call:

CALL LDFLOC (NR, NNAM, NAMES,X)

The argument list description

NR — this integer variable receives a returncode, which is set by the subroutine, indication whether or not data are found

= 0 no data found for requested data type

= 1 requested data are available, first data item is stored in the array X

NNAM(1)	number of names for the specified data type		filled by LDFLOC
NNAM(2)	number of arguments	for a single data item	
NNAM(3)	number of functional values		

see section 1.3.1 and Ref. 1 for explanation

NAMES is a real array of adequate length to retain the names for the requested data type. These names must be assigned before the call. They may be altered by the subroutine.

X is an array of adequate length to retain the first data item for the requested type. It is filled by the LDFLOC routine.

NOTE: Example for modification of names by LDFLOC: e.g. for the inelastic excitation the next name greater or equal to the given name is returned. In general: The further names (floating point notation)are modified to the first set of names greater than or equal to the given one. The order of precedence is the same as that one defined in KEDAK (see ref. 1, see section 1.3.2).
LDFLOC locates data for alphameric names, the ENTRY IDFLOC (NR, NNAM, NAMES,X) performs the same function for numeric names (arguments see LDFLOC).LDFLOC and IDFLOC cen be (and normally are) called by the user.

The subroutine LDFLOC has two additional entries reserved for internal use only:

ENTRY LDFTTO (NR, NAMES, IDF)
called by LDFITN, LDFNAM (3.2.4, 3.2.5)

ENTRY LDFRQO (NR, NAMES, IDF)
called by LDFDPN (3.3.10)

A call to the LDFTTO entry provides the type address table (see reference 1). This table is loaded from the KEDAK-file if it is not yet available in the LDFTT common block from an earlier call. A call to the LDFRQO entry provides the contents of the LDFRQ common (see 3.1.3).

The arguments:

NR, NAMES are the same as described for the LDFLOC entry

IDF is an integer variable indicating whether the names are alphameric or numeric.

= 1 numeric names
= 2 alphameric names

Externals used

The LDFLOC routine calls the LDFROP-Entry of LDFOPN (3.2.1) to reload the tables from the KEDAK-file if the information in the common blocks LDFMT, LDFTT (3.1.1, 3.1.2) has been destroyed (e.g. overlayed). If the currently requested data type does not belong to the same material as the previously requested one, a call to the LDFTYP routine (3.3.6) provides the type address-tables in the common LDFTT and a subsequently call to LDFTYC (3.3.9) provides the pointers to the data block for the requested type in the LDFRQ common (3.1.3).

If the requested data type has further names the pointers received by LDFTYC do not point to the data block but to the further names address table. A call to LDFFNA (3.3.7) provides the pointers to the data block for this type with further names.

If more than one KEDAK-file is in use and if the requested material is not on the file currently used,a call to LDFALF provides the material address table from an alternative file in the common LDFMT (see 3.1.1). ATTENTION: If a material is not found on the current file it is searched for on all other files opened by the program,until it is either found or all files have been searched.

If more than one file has been opened by the program,the file currently linked to the program when returning from LDFLOC has not to be the same one as the file at the time of the call.

The program size of the LDFLOC routine is 2010 bytes when using the H-extended compiler on IBM 370/168.

3.2.3 The subroutine LDFNXT

The LDFNXT subroutine retrieves the next data item (see 2.1) from the data block. The data items in the block are stored in ascending order of arguments if the number of arguments is one (otherwise the order of precedence as defined in ref. 1). A call to the LDFLOC routine (3.2.2) must precede the first LDFNXT call and provides the first data item. For each next data item a call to LDFNXT has to be performed. The return code NR is set to one if there are available more data for requested data type in the data block. If no more data are available return code NR is set to zero. Return code setting to zero is also done in the case of further names if no more data items for a given name combination can be found. But if there exists another further name combination, these further names are loaded in NAMES and the first data item is stored in X.

The call:

CALL LDFNXT (NR, NNAM, NAMES, X)

LDFNXT retrieves data for alphameric names. The ENTRY IDFNXT (NR, NNAM, NAMES, X) performs the same function for numeric names.

The arguments for LDFNXT and IDFNXT are the same as described in the LDFLOC routine (see 3.2.2).

Externals used:

The LDFREC routine (3.3.8) is called to read a record from the KEDAK-file. The labeled common blocks: LDFRC, LDFRQ, LDFIL and LDFRE are used.

The program size is 1730 bytes.

3.2.4 Subroutine LDFITN

The LDFITN subroutine provides the names of the data types available on the KEDAK-file for the specified material MAT. These names are stored into the ITYPES array.

The call:

```
CALL LDFITN (NR, MAT, ITYPES, NUM, MAX, NMAX, IDF)
```

The argument list description

NR — is an integer variable filled by the subroutine with a return code indicating the result of the call

= 1 no error, call was successful

= 0 for MAX = 0: error in LDFOPN or MAT not found on the file.

= 0 for MAX $\neq$ 0: number of data types exceeds the length of array ITYPES

MAT — is a variable to provide the material name. It must be assigned before the call (REAL*8).

ITYPES — is an array to retain the names of data types available for the material MAT (REAL*8).

NUM — is an integer array to retain the number of data points for each data type available. If the type has further names, NUM reflects the number of name combinations for that type, e.g. for 'SGNC' the number of (elastic scattering angular) distributions.

MAX — is an integer variable that indicates the number of data types available for the material MAT.

NOTE: ITYPES, NUM and MAX are filled by the LDFITN routine.

NMAX — is an integer variable that specifies the length of the ITYPES and NUM arrays.

IDF — is an integer variable that indicates whether the names are numerical or alphamerical as follows:

= 1 numeric names
= 2 alphameric names.

NOTE: MMAX and IDF must be assigned before the call.

Externals used:

The LDFITN subroutine calls the LDFTTO entry in LDFLOC (see 3.2.2) to provide information from type address table and uses the following two common blocks: LDFTT and LDFTC. The numeric type names are stored into ITYP area of LDFTC. The program size is 982 bytes main storage on the IBM 370/168 with the H-extended compiler.

3.2.5 Subroutine LDFNAM

The subroutine LDFNAM provides information from the address table of further names (see 1.3.1 and 1.3.2). For a data type with more than two names all numeric further names are stored into the NAMX area.

The call:

```
CALL LDFNAM (NR, NAMES, XNAM, N1, N2, MAX, IDF)
```

The argument list description:

NR — is an integer variable filled by the subroutine with a return code indicating the result of the call:

= 1 no error, call was successful
= 0 for N2 > 0 error, indicates that the
number of further names is greater than MAX, the
maximal length of the XNAM array
= 0 for N1 = N2 = 0: no further names for requested
data type

NAMES an array that gives the name combination for which the further names are to be supplied, e.g.: 'U235', 'SGIZ' for IDF = 2. The name combination must be assigned before the call.

XNAM is an array of adequate length to receive the numeric further names stored by the LDFNAM routine.

MAX is an integer variable that specifies the dimension of XNAM. It has to be assigned before the call.

N1 is an integer variable that specifies the number of numeric names for one name combination, e.g.: for 'SGIZ': N1 = 1, for'LEGNC' : N1 = 2

N2 is an integer variable that specifies the number of available name combinations. N1 and N2 are ascertained and stored by the LDFNAM routine.

IDF is an integer variable that specifies whether alphameric or numerical names are used in the call statement as follows:

= 1 numerical names
= 2 alphamerical names

It must be assigned before the call.

Externals used:

The LDFNAM subroutine calls the LDFTTO entry of the LDFLOC routine (see 3.2.2) to provide the type addresstable, the LDFTYC routine (see 3.3.9) to fetch the pointers to the types and the LDFREC routine (see 3.3.8) to read one record from the KEDAK-file.

The following labeled common blocks are used by the LDFNAM subroutine: /LDFTT/, /LDFRQ/, /LDFRC/, /LDFIL/, /LDFRE/.

The program size is 1050 bytes.

3.2.6 Subroutine LDFERR

The subroutine LDFERR prints the error diagnostics for the retrieval packages IDFPAC or LDFPAC. The message number, material and type name for the message are given in the common /LDFRC/ (see 3.1.7). LDFERR has no arguments. If an error occurs during processing the retrieval routines, the error number is assigned to the error indicator NRETCD in /LDFRC/. A call for LDFERR effects a check of the error indicator, and the print out of the corresponding message.

Use of LDFERR

A call to a subroutine of the retrieval packages IDFPAC or LDFPAC was not successful, i.e. the return code NR indicates an error. Then the user may obtain the error diagnostic message for this error by a call to LDFERR.

For instance:

```
      .
      .
      .
      COMMON /INOUT/KOUT
      COMMON /LDFIL/IFIL
      .
      .
      .
      CALL LDFOPN (IFIL, IDAT, &999)
      .
      .
      .
C     ERROR IN LDFOPN - PRINT ERROR MESSAGE

999   CALL LDFERR
      STOP
```

The printing unit is specified by KOUT in the common INOUT. The file number of the KEDAK-file is indicated by IFIL in the common LDFIL.

Arguments: non

3.3 Detailed description of the auxiliary subroutines

3.3.1 Subroutine LDFTB1

The LDFTB1 subroutine loades information from the material conversion table, type conversion table, and the material address table into the common blocks /LDFTC/, /LDFMT/, and uses the material conversion table to create an alphameric material address table.

The call:

CALL LDFTB1 (IDAT, IFL)

The arguments:

IDAT is an integer variable, which is filled with the creation date of the current file

IFL is an integer variable that indicates the data set reference number of the requested file. It must be assigned before the call.

Externals used:

The LDFCTB routine (see 3.3.2) is called twice by LDFTB1, to retrieve the material and the type conversion tables. A call to the LDFMAT routine (see 3.3.4) provides the material address table and a call to the LDFMAC routine (see 3.3.3) the alphameric material address table.

The common blocks /LDFTC/ and /LDFMT/ are filled by LDFTB1, the common /LDFREC/ is changed, and /LDFIL/ is used.

The program size is 169 k bytes main storage on 370/168 with the H-extended Compiler.

3.3.2 Subroutine LDFCTB

The LDFCTB subroutine retrieves the type or material names conversion table (see reference 1 and Appendix 1) from the KEDAK-file.

CALL LDFCTB (IX, LT, IWT, IT)

The argument list description:

IX is an integer array with two elements specifying the pointers to the conversion tables: the record number, and the word number. The conversion table starting at this address is read into the two arrays supplied by the call.

LT is an integer variable that specifies the number of names in the conversion table. IX, LT are obtained from the first record of the KEDAK-file for each of the two conversion tables.

IWT is an array to retain the alphameric names.

IT is an array to retain the numeric names.

LDFCTB is for internal use only, it is called by LDFTB1.

Externals used

The common blocks /LDFRE/ and /LDFIL/ are used in the LDFCTB routine, and the LDFREC subroutine (see 3.3.8) is called to read a record from the KEDAK-file.

Storage requirements: 578 bytes with the H-extended compiler on 370/168.

3.3.3 The Subroutine LDFMAC

The LDFMAC subroutine converts the material address table from numeric to alphameric material names and provides them in the M2 area. LDFMAC is called by LDFTB1 (3.3.1). LDFMAC is for internal use only.

```
CALL LDFMAC (M1, I1, L1, M2, I2, L2)
```

The argument list description,

M1 is a REAL*8 array containing the alphameric material names from the material conversion table, it is filled before the call.

I1 is an integer array containing the numeric material names.

L1 is an integer variable that indicates the number of materials in the conversion table.

M2 is a REAL*8 array to receive the alphameric names of the materials available on the KEDAK-file.

I2 is an integer array giving the numeric names of the materials available on the KEDAK-file.

L2 is an integer variable that indicates the number of materials available.

NOTE: All arguments, with the exception of M2, must be assigned before the call.

Storage requirements: 566 bytes main storage with the H-extended compiler on IBM 370/168.

3.3.4 Subroutine LDFMAT

The subroutine LDFMAT retrieves the material address table (see Appendix 1, reference 1) from the KEDAK-file. LDFMAT is for internal use only and is called by the subroutine LDFTB1 (see 3.3.1).

The call:

```
CALL LDFMAT (IX, LX, IT, NT, AT)
```

The argument list description

IX — is an integer array with two elements containing the pointers (record number and word number) to the table of materials available on the KEDAK-file ("Material address table").

LX — is an integer variable that indicates the number of materials on the KEDAK-file.

<u>NOTE:</u> IX, LX are obtained from the first record of the KEDAK-file, they must be filled before the call

IT — is an integer array to store the numeric material names.

NT — is an INTEGER*2 array to receive the number of reaction types for each material.

AT — is an INTEGER*2 array to receive the pointers (record number and word number) to the type address table for each material.

Externals used:

The LDFMAT routine uses and alters the common blocks /LDFRC/,/LDFRE/,/LDFIL/ and calls the LDFREC routine to read a record from the KEDAK-file. The Program size is 816 bytes.

3.3.5 Subroutine LDFTAT

The LDFTAT subroutine loads the type address table from the declaration part of the KEDAK-file into the LDFTT common block. The array ITYNAM is filled with the type names and the ITYTAB-array with the pointers from the type address table.

The call:

```
CALL LDFTAT (LTYP)
```

The argument LTYP is an integer variable that specifies the number of types available for the requested material on the KEDAK-file.

Externals used:

The LDFREC subroutine is called to read a record from the KEDAK-file, and the following common blocks are used: /LDFTT/, /LDFRC/, /LDFIL/, /LDFRE/.

Storage requirements for the LDFTAT subroutine: the program size is 584 bytes.

3.3.6 Subroutine LDFTYP

The LDFTYP subroutine retrieves the pointers to the type address table for the requested material MAT, and calls LDFTAT (3.3.5) to load the type address table. LDFTYP is for internal use only, called by LDFLOC.

The call:

CALL LDFTYP (IDF)

The argument IDF is an integer variable that indicates, whether the names are numeric or alphameric: 1 - numeric names, 2 - alphanumeric names.

Externals used:

The LDFTAT subroutine is called to load the type address table from the KEDAK-file.

The common blocks LDFRE, LDFIL and LDFRC are used.

Storage requirements for the LDFTYP routine: program size is 590 bytes.

3.3.7 The subroutine LDFFNA

The LDFFNA subroutine provides information in case of data types of mor than two names. The pointers to the data array are usually stored in NAM(4-6) of the common /LDFRQ/ (3.1.3). If the data type has further names, NAM (4-6) contain the pointers to the further name address table. LDFFNA stores these pointers into NFNAM (1-3); by using them, LDFFNA retrieves the pointers to the data block for the requested name combination from the further name address table and stores them into NAM (4-6) for later use in the LDFNXT routine.

If the end of data for one name combination is reached, LDFNXT uses the information in NFNAM (1-3) and in NFDONE (common /LDFRQ/) to retrieve (from the further name address table) the pointers to the data block of the next name combination. By this it is not necessary to search for the needed pointer the whole address table of further names. LDFFNA is for internal use only, called by LDFLOC.

The call:

CALL LDFFNA (NAMES, CHNT, IDF)

The argument list description:

NAMES	is an array giving the names of the requested data type and must be filled prior to the call.
CHNT	is a logical variable that indicates whether or not NAM (4-6) in the LDFRQ common are to be stored into NFNAM (1-3).
IDF	is an integer variable that indicates whether or not names are alphameric or numeric as follows: = 1 numeric names = 2 alphameric names

Externals used:

The LDFREC subroutine is called to read one record from the KEDAK-file. The common blocks /LDFRQ/ and /LDFREC/ are used and partially altered.

Storage requirements for the LDFFNA routine on IBM 370/168 with the H-extended compiler: program size is 1072 bytes.

3.3.8 Subroutine LDFREC

The LDFREC routine reads one record from the direct access KEDAK-dataset. The common blocks /LDFRE/ and /LDFIL/ are used. The program size is 232 bytes with the H-extended compiler on IBM 370/168.

Arguments: none.

3.3.9 Subroutine LDFTYC

The subroutine LDFTYC retrieves the pointers for the requested data type. LDFTYC is called by the subroutines LDFLOC (3.2.2) and LDFNAM (3.2.5) and is for internal use only. The type address table stored in the common /LDFTT/ is searched for the requested type. After the requested type has been found the pointers are copied from the array ITYTAB in the common block /LDFTT/ into the array NAMX in the common block /LDFRQ/.

The call:

CALL LDFTYC (IDF)

The argument IDF is an integer variable that indicates whether the names are numeric or alphameric as follows:

1 - integer, 2 - alphameric

Externals used:

The following common blocks are used and eventually updated: LDFTT, LDFTC, LDFRC, LDFRQ.

Storage requirements for the LDFTYC routine: The program size is 916 bytes on 370/186 with the H-extended compiler.

3.3.10 Subroutine LDFDPN

The LDFDPN subroutine provides the number of data points for a given name combination. A call to the LDFRQ entry in the subroutine LDFLOC (see 3.2.2) loads the common block LDFRQ which is used by the LDFDPN routine. The program size is 356 bytes on IBM 370/168 with the H-extended compiler.

The call:

CALL LDFDPN (NR, NAMES, NUM, IDF)

Argument list description:

NR — is an integer variable which is filled by the subroutine with a returncode indicating the result of the call
= 0 error
= 1 no error

NAMES is an array that specifies the name combination for which the number of data points is enquired.

NUM is an integer variable that returns the number of data points.

IDF is an integer variable indicating whether the given names are numerical or alphamerical as follows:

= 1 - numeric names
= 2 - alphameric names

NOTE: NAMES and IDF must be assigned before the call.

References

1. B. Krieg
Handling and Service programs for the Karlsruhe Nuclear Data File KEDAK
Institut für Neutronenphysik und Reaktortechnik, Gesellschaft für Kernforschung mbH., KFK 1725, Juni 1973 (or KFK 2387/II)

2. I. Langner
Anpassung von SCORE an KEDAK und NEUDADA-Daten. SCORE-Version-III-K-1972.
Not published

3. G. Arnecke, H. Bachmann
DINF und DEFI, Programme zur Dynamisierung des DEFINE FILE statements.
Not published

4. R. Meyer
RETPAC, a user oriented retrieval package for use with the Evaluated Nuclear Data Library KEDAK.
Institut für Neutronenphysik und Reaktortechnik, Gesellschaft für Kernforschung mbH., KFK 2387/III.4

5. IBM SYSTEM / 360
Operating System Linkage Editor
Form C28-65 38

6. I. Langner, R. Meyer
Description for CALCUL.
KFK-report in preparation.

7. C. Broeders
Description for PLKFG.
Institut für Neutronenphysik und Reaktortechnik, Gesellschaft für Kernforschung mbH., KFK 2387/IV.3.1

8. I. Langner, R. Meyer
Programm zum Umkopieren der von Brigitte erstellten ENDF/B-Datei nach KNDF.
Institut für Neutronenphysik und Reaktortechnik, Gesellschaft für Kernforschung mbH., KFK 2387/VI.3

Acknowledgement

The authors are indebted to Mr. Goel and Mr. Stein for their critical reading of this report.

5. Source lists for IDFPAC

Subroutine name	Page
LDFERR	III-71
LDFNAM	III-73
LDFITN	III-74
LDFDPN	III-75
LDFOPN	III-75
LDFLOC	III-76
LDFNXT	III-78
LDFCTB	III-80
LDFREC	III-81
LDFMAT	III-81
LDFFNA	III-82
LDFMAC	III-83
LDFTB1	III-83
LDFTYC	III-84
LDFTAT	III-85
LDFTYP	III-86

```
      SUBROUTINE LDFERR
      COMMON/INOUT/KOUT
      COMMON/LDFRC/NRETCD,NXF,MAT,TYP,XF(4)
      COMMON/LDFIL/IFIL
      DIMENSION MAT(3),TYP(3)
C
C          PRINT ERRORDIAGNOSTIC.
C         KOUT.....PRINTING UNIT.
      NERR=0
      WRITE(KOUT,600)
  600 FORMAT(/T5,'ERRORDIAGNOSTIC FOR KEDAKRETRIEVAL:'/)
      IF(NRETCD.EQ.0) GOTO 910
      GOTO (1,2,3,4,5,6,7,8,9,10,11,12,13,14,15,16,17),NRETCD
      GOTO 900
    1 WRITE(KOUT,601) IFIL
  601 FORMAT(T10,'DATASET ON UNIT:',I2,' DOES NOT CONTAIN CORRECT KEDAK-
     1IDENTIFICATION'/)
      GOTO 1000
    2 WRITE(KOUT,602) IFIL
  602 FORMAT(T10,'LENGTH OF MATERIAL CONVERSION TABLE ON UNIT:',I2,' EXC
     1EEDS AVAILABLE STORAGE.CALL PROGRAMMER.'/)
      GOTO 1000
    3 WRITE(KOUT,603) IFIL
  603 FORMAT(T10,'LENGTH OF TYPE CONVERSION TABLE ON UNIT:',I2,' EXCEEDS
     1 AVAILABLE STORAGE.CALL PROGRAMMER.'/)
      GOTO 1000
    4 WRITE(KOUT,604) IFIL
  604 FORMAT(T10,'LENGTH OF MATERIAL ADRESS TABLE ON UNIT:',I2,' EXCEEDS
     1 AVAILABLE STORAGE.CALL PROGRAMMER.'/)
      GOTO 1000
    5 WRITE(KOUT,605)
  605 FORMAT(T10,'FILE NUMBER USED IN LDFOPN IS OUT OF VALID RANGE.'/)
      GOTO 1000
    6 WRITE(KOUT,606)
  606 FORMAT(T10,'LDFLOC CALL WAS EXECUTED,BUT NO FILE WAS OPENED BEFORE
     1.'/)
      GOTO 1000
    7 WRITE(KOUT,607)MAT
  607 FORMAT(T10,'REQUESTED MATERIAL:',2A4,' OR ',I8,' NOT ON AVAILABLE
     1FILES.'/)
      GOTO 920
    8 WRITE(KOUT,608) IFIL
  608 FORMAT(T10,'FIXPOINT OVERFLOW IN LENGTH-ITEM OF MATERIAL-ADRESSTAB
     1LE ON UNIT:',I2,'.CALL PROGRAMMER.'/)
      GOTO 1000
    9 WRITE(KOUT,609) IFIL
  609 FORMAT(T10,'FIXPOINT OVERFLOW IN ADRESS-ITEM OF MATERIAL-ADRESSTAB
     1LE ON UNIT:',I2,'.CALL PROGRAMMER.'/)
      GOTO 1000
   10 WRITE(KOUT,610) MAT
  610 FORMAT( T10,'LENGTH OF TYPE-ADRESSTABLE FOR ',2A4,' OR ',I8,' EXCE
     1EDS AVAILABLE STORAGE.CALL PROGRAMMER.'/)
      GOTO 1000
   11 WRITE(KOUT,611) MAT
  611 FORMAT( T10,'FIXPOINT OVERFLOW IN WORDS 2-7 OF TYPE-ADRESS TABLE F
     1OR ',2A4,' OR ',I8,'.CALL PROGRAMMER.'/)
      GOTO 1000
   12 WRITE(KOUT,612) IFIL,TYP
  612 FORMAT( T10,'TYPE-CONVERSIONTABLE ON UNIT:',I2,' DOES NOT CONTAIN
     1',2A4,' OR ',I8,'.'/)
      GOTO 920
   13 WRITE(KOUT,613) MAT,IFIL,TYP
  613 FORMAT(T10,'MATERIAL ',2A4,' OR ',I8,' ON UNIT ',I2,' DOES NOT CON
     1TAIN TYPE ',2A4,' OR ',I8,'.'/)
      GOTO 1000
   14 WRITE(KOUT,614) MAT,TYP,IFIL
  614 FORMAT(T10,'NUMBER OF FURTHER NAMES FOR MATERIAL:',2A4,' OR ',I8,'
     1,TYPE:',2A4,' OR ',I8,',ON UNIT ',I2,' EXCEEDS AVAILABLE STORAGE.'
     2//T25,'CALL PROGRAMMER.'/)
      GOTO 1000
   15 WRITE(KOUT,615) MAT,TYP,IFIL,(XF(I),I=1,NXF)
  615 FORMAT(T10,'FOR MATERIAL:',2A4,' OR ',I8,',TYPE:',2A4,' OR ',I8,',
     1 ON UNIT ',I2,' AT LEAST ONE OF THE REQUESTED NUMERICAL NAMES'//
     2T10,'IS ABOVE THE LARGEST ONE STORED ON KEDAK.'//
     3T10,'THE NUMERICAL NAMES REQUESTED ARE :',1P4E13.5)
      WRITE(KOUT,903)
  903 FORMAT(1X)
      GOTO 1000
   16 WRITE(KOUT,616) MAT,TYP
      IF(NXF.NE.0) WRITE(KOUT,902) (XF(I),I=1,NXF)
  616 FORMAT(T10,'LDFNXT WAS CALLED WITHOUT PREVIOUS SUCCESSFUL EXECUTIO
     1N OF LDFLOC FOR:'//
     2 T20,'MATERIAL:',2A4,' OR ',I8,',     TYPE:',2A4,' OR ',I8/)
  902 FORMAT(T20,'NUMERIC NAMES (IF ANY):',1P4E13.5)
      WRITE(KOUT,903)
      GOTO 920
   17 WRITE(KOUT,617)
  617 FORMAT(T10,'ATTEMPT TO OPEN A NEW FILE BY CALL TO LDFOPN,THOUGH MA
     1XIMUM NUMBER HAS ALREADY BEEN OPENED.'/)
      GOTO 1000
  900 WRITE(KOUT,901) NRETCD
  901 FORMAT(T10,'NO MESSAGE STORED FOR THIS ERRORTYPE.RETURNCODE WAS:',
     1I3/)
      GOTO 1000
  910 WRITE(KOUT,911)
  911 FORMAT(T10,'NO ERRORCONDITION OCURRED.'/)
      GOTO 1000
  920 WRITE(KOUT,921)
  921 FORMAT(T10,'ONLY ONE OF BOTH NAME-REPRESENTATIONS IN MESSAGETEXT N
     1EED BE VALID.'/)
      GOTO 1000
 1000 RETURN
      END
      SUBROUTINE LDFNAM(NR,NAMES,XNAM,N1,N2,MAX,IDF)
      DIMENSION XNAM(1),MAT(3),TYP(3),NAMES(1),IST(2)
      REAL*8 ID
      REAL MAT,NAMES
      COMMON/LDFTT/TMAT(3)
      COMMON/LDFRQ/MAT,TYP,FNAM(4),NPO(6)
      COMMON/LDFRC/NRC
      COMMON/LDFIL/IFIL,LREC,LR
      COMMON/LDFRE/ID,W(880),LW
      DATA IST/3,1/
C
C PROVIDE INFORMATION FROM ADRESSTABLE OF FURTHER NAMES.
C XNAM:FURTHER NAMES,N1:NUMBER/COMBINATION,N2:NUMBER OF COMBINATIONS.
C         NR...RETURN CODE: 0=ERROR,1=NO ERROR.
C                NR=0,N2>0 INDICATES OVERFLOW OF XNAM.
C         NAMES....NAME COMBINATION.INTEGER IF IDF=1.REAL*8 IF IDF=2.
      N1=0
      N2=0
      CALL LDFTTO(NR,NAMES,IDF)
      IF(NR.EQ.0) GOTO 100
      NR=0
      J=IST(IDF)
      DO 2 I=1,IDF
      TYP(J)=NAMES(IDF+I)
    2 J=J+1
      DO 3 I=1,3
    3 MAT(I)=TMAT(I)
      CALL LDFTYC(IDF)
      IF(NRC.NE.0) GOTO 100
      DO 5 I=1,4
    5 FNAM(I)=0
      N1=NPO(1)
      IF(N1.EQ.0) GOTO 100
      LR=NPO(5)
      READ(IFIL'LR) W
      LW=NPO(6)-1
      NX=NPO(4)
      DO 10 I=1,NX
      DO 20 J=1,N1
      N2=N2+1
      IF(N2.GT.MAX) GOTO 90
      LW=LW+1
      IF(LW.GT.LREC) CALL LDFREC
   20 XNAM(N2)=W(LW)
      DO 30 J=1,3
      LW=LW+1
      IF(LW.GT.LREC) CALL LDFREC
   30 CONTINUE
   10 CONTINUE
      NR=1
      GOTO 92
   90 N2=N2-1
   92 N2=N2/N1
  100 DO 102 I=1,3
  102 TYP(I)=0
```

```
      RETURN
      END




      SUBROUTINE LDFITN(NR,MAT,ITYPES,NUM,MAX,MMAX,IDF)
      DIMENSION NUM(1),ITYPES(1),MAT(1)
      COMMON/LDFTT/MO(3),ITYNAM(70),ITYTAB(6,70),LTYTAB
      INTEGER*2 ITYTAB
      COMMON/LDFTC/DUM(4),ITYPS(140),ITYP(70),LTYCON
      INTEGER BL/'    '/
C
C          PROVIDE SOME INFORMATION FROM TYPE-ADRESSTABLE.
C        MAT ... MATERIAL NAME. INTEGER IF IDF=1. REAL*8 IF IDF=2.
C        ITYPES....TYPENAMES OF AVAILABLE TYPES FOR THAT MATERIAL.
C              INTEGER IF IDF=1. REAL*8 IF IDF=2.
C        NUM(I) NUMBER OF DATAPOINTS FOR I-TH TYPE.IF THIS TYPE HAS
C              NUMERIC NAMES ALSO,NUM(I) IS THE NUMBER OF 0
C              NUMERIC NAMES ALSO,NUM(I) IS THE NUMBER OF COMBINATIONS
C              FOR THAT TYPE,E.G. FOR SGNC THE NUMBER OF DISTRIBUTIONS.
C        MAX NUMBER OF AVAILABLE TYPES.
C        MMAX LENGTH OF ARRAYS ITYPES(IN R*8 OR I*4 RESP.) AND NUM(I*4).
C        NR....RETURNCODE.0=ERROR. 1=NO ERROR.
C              NR=0,MAX>0 INDICATES OVERFLOW OF ARRAYS ITYPES,NUM.
C
      CALL LDFTTO(NR,MAT,IDF)
      IF(NR.EQ.0) GOTO 110
      MAX=LTYTAB
      IF(MAX.GT.MMAX) MAX=MMAX
      IF(IDF.EQ.1) GOTO 200
      DO 100 L=1,LTYTAB
      IF(L.GT.MMAX) GOTO 120
      DO 10 I=1,LTYCON
      IF(ITYP(I).NE.ITYNAM(L)) GOTO 10
      ITYPES(2*L-1)=ITYPS(2*I-1)
      ITYPES(2*L)=ITYPS(2*I)
      NUM(L)=ITYTAB(4,L)
      GOTO 100
   10 CONTINUE
      ITYPES(2*L-1)=BL
      ITYPES(2*L)=BL
      NUM(L)=0
  100 CONTINUE
      NR=1
      GOTO 1000
  110 MAX=0
      GOTO 1000
  120 NR=0
      GOTO 1000
  200 DO 210 L=1,LTYTAB
      IF(L.GT.MMAX) GOTO 120
      ITYPES(L)=ITYNAM(L)
  210 NUM(L)=ITYTAB(4,L)
      NR=1
      GOTO 1000
 1000 RETURN
      END




      SUBROUTINE LDFDPN(NR,NAMES,NUM,IDF)
      DIMENSION NNAM(3),NAMES(1)
      COMMON/LDFRQ/XDUM(13),NDP
C
C         PROVIDE NUMBER OF DATAPOINTS FOR GIVEN NAMECOMBINATION.
C        NAMES....NAMECOMBINATION. INTEGER IF IDF=1. REAL*8 IF IDF=2.
C        NUM....NUMBER OF DATAPOINTS.
C        NR....RETURNCODE: 0=ERROR, 1=NO ERROR.
      NUM=0
      CALL LDFRQ0(NR,NAMES,IDF)
      IF(NR.EQ.0) GOTO 10
      NUM=NDP
      NR=1
   10 RETURN
      END




      SUBROUTINE LDFOPN(IFI,IDAT,*)
      COMMON/LDFRC/NRETCD
      COMMON/LDFIL/IFIL,LRECD,LR
      DIMENSION NFIL(5)
      DATA IFL/0/,LREC/880/,MAXFIL/5/,NFIL/5*0/,LDS/8000/
      DATA XU/'U'/
C
C  LDFOPN:DEFINES THE FILE,PROVIDES FIRST THREE TABLES AND KEEPS
C         SOME INFORMATION.IT MUST NOT BE OVERLAYED.
      NRETCD=0
      IF(IFI.LE.0.OR.IFI.GT.99) GOTO 90
      DO 2 I=1,MAXFIL
      IF(NFIL(I).EQ.IFI) GOTO 1
      IF(NFIL(I).NE.0) GOTO 2
      IFL=I
      NFIL(I)=IFI
      GOTO 4
    1 IFL=I
      GOTO 20
    2 CONTINUE
      NRETCD=17
      GOTO 100
    4 CALL DEFI(IFI,LDS,XU,LREC,LR)
      IF(LDS.GE.0) GOTO 20
      LDS=-LDS
      GOTO 90
C
C         REREAD TABLES OF CURRENT FILE.
      ENTRY LDFROP(IFI,*)
      IF(IFL.NE.0) GOTO 15
   12 NRETCD=6
      GOTO 100
   15 IFI=NFIL(IFL)
   20 IFIL=IFI
      LRECD=LREC
      CALL LDFTBI(IDAT,IFL)
      IF(NRETCD.NE.0) GOTO 100
   50 RETURN
   90 NRETCD=5
  100 RETURN 1
C
C          LOAD TABLES OF AN ALTERNATIVE FILE.
      ENTRY LDFALF(IFS,IFI,*)
      IF(IFL.EQ.0) GOTO 12
      IFLK=IFL
      IFL=IFL+1
      IF(IFL.GT.MAXFIL) IFL=1
      IF(NFIL(IFL).EQ.0) IFL=1
      IFI=NFIL(IFL)
      IF(IFI.NE.IFS) GOTO 20
      IFL=IFLK
      NRETCD=7
      GOTO 100
      END




      SUBROUTINE LDFLOC(NR,NNAM,NAMES, X)
      DIMENSION NNAM(1),NAMES(1),X(1)
      INTEGER X, IST(2)/3, 1/
      REAL NAMES,MAT(3),XM(3),TYP(3),XT(3),MAT1(3)
      REAL*8 IDZ,IDENT/'KEDABIBL'/
      INTEGER*2 ITYTAB
      LOGICAL TTONLY,NODATA,CHNT
      COMMON/LDFTT/MAT,ITYNAM(70),ITYTAB(6,70),LTYTAB
      COMMON/LDFRC/NRETCD,IDUM,XM,XT,XF(4)
      COMMON/LDFRF/IDZ,IW(880),LW
      COMMON/LDFIL/IFIL,LRECD,LR
      COMMON/LDFRQ/MAT1,TYP,FNAM(4),NAM,NAMX(8),NRDONE,NFDONE
      DATA MFNAM/4/
C
C          LDFLOC: LOCATE DATA OF REQUESTED NAMES.
C                  PROVIDE FIRST DATA POINT.
      IDF=2
    2 TTONLY=.FALSE.
      NODATA=.FALSE.
      GOTO 10
      ENTRY IDFLOC(NR,NNAM,NAMES,X)
      IDF=1
      GOTO 2
C
C          PROVIDE TYPE-ADRESSTABLE ONLY.
      ENTRY LDFTTO(NR,NAMES,IDF)
      TTONLY=.TRUE.
      GOTO 10
C
C         PROVIDE CONTENTS OF LDFRQ ONLY.
      ENTRY LDFRQO(NR,NAMES,IDF)
      TTONLY=.FALSE.
      NODATA=.TRUE.
      GOTO 10
C
C         REDO LDFOPN,IF NECESSARY.
   10 CALL LDFROP(IFIL,&999)
C
C         LOAD TYPE-ADRESSTABLE IF NECESSARY.
      NUMX=IST(IDF)
      DO 11 I=1,IDF
      IF(MAT(NUMX).NE.NAMES(I).OR.MAT1(NUMX).NE.NAMES(I)) GOTO 13
   11 NUMX=NUMX+1
      GOTO 30
   13 NUMX=IST(IDF)
      DO 14 I=1,IDF
      IF(MAT(NUMX).NE.NAMES(I)) GOTO 15
   14 NUMX=NUMX+1
      GOTO 32
   15 I=IFIL
   12 NUMX=IST(IDF)
      DO 16 J=1,IDF
      MAT(NUMX)=NAMES(J)
   16 NUMX=NUMX+1
      CALL LDFTYP(IDF)
      IF(NRETCD.EQ.7) GOTO 20
      IF(NRETCD.NE.0) GOTO 999
      GOTO 32
C
C         MATERIAL NOT ON CURRENT UNIT.TRY ALTERNATIVE UNIT.
   20 CALL LDFALF(I,IFIL,&999)
      GOTO 12
C
C         FETCH POINTERS TO REQUESTED TYPE,IF NECESSARY.
   30 IF(TTONLY)  GOTO 1001
      NUMX=IST(IDF)
      DO 35 I=1,IDF
      IF(TYP(NUMX).NE.NAMES(IDF+I)) GOTO 33
   35 NUMX=NUMX+1
      GOTO 40
   32 IF(TTONLY) GOTO 1001
   33 NUMX=IST(IDF)
      DO 36 I=1,IDF
      TYP(NUMX)=NAMES(IDF+I)
   36 NUMX=NUMX+1
      DO 37 I=1,3
   37 MAT1(I)=MAT(I)
      CALL LDFTYC(IDF)
      IF(NRETCD.NE.0) GOTO 999
      IF(NAM.EQ.0) GOTO 90
      IF(NAM.GT.MFNAM) GOTO 42
      CHNT=.FALSE.
      GOTO 60
C
C           FURTHER NAMES?
   40 IF(NAM.EQ.0) GOTO 90
      IF(NAM.GT.MFNAM) GOTO 42
      CHNT=.TRUE.
      GOTO 60
   42 NRETCD=14
      DO 43 I=1,3
      XM(I)=MAT(I)
   43 XT(I)=TYP(I)
      GOTO 999
   60 CALL LDFFNA(NAMES,CHNT,IDF)
      IF(NRETCD.NE.0) GOTO 999
C
C           FILL NNAM.
   90 NRDONE=0
      IF(NODATA) GOTO 1001
      NNAM(1)=NAM+2
      NNAM(2)=NAMX(1)
      NNAM(3)=NAMX(2)
C
C           FETCH FIRST RECORD.
      LR=NAMX(4)
      LW=NAMX(5)-1
      READ(IFIL'LR) IW
      IDZ=IDENT
      NUMX=NAMX(1)+NAMX(2)
      DO 92 I=1,NUMX
      LW=LW+1
      IF(LW.GT.LRECD) CALL  LDFREC
   92 X(I)=IW(LW)
      NRDONE=1
 1001 NR=1
      GOTO 1000
  999 NR=0
 1000 RETURN
      END




      SUBROUTINE LDFNXT(NR,NNAM,NAMES,X)
      DIMENSION NAMES(1),X(1),NNAM(1)
      INTEGER IST(2)/3,1/
      REAL NAMES,MAT(3),TYP(3),XM(3),XT(3)
      REAL*8 IDZ,IDENT/'KEDABIBL'/
      COMMON/LDFRE/IDZ,W(880),LW
      COMMON/LDFRC/NRETCD,INAM,XM,XT,XF(4)
      COMMON/LDFRQ/MAT,TYP,FURNAM(4),N1(6),N2(3),NP,NFN
      COMMON/LDFIL/IFIL,LREC,LR
      INTEGER IW(880)
      EQUIVALENCE (IW(1),W(1))
C
```

```
C          RETRIEVE NEXT DATAPOINT.
      IDF=2
      GOTO 2
      ENTRY IDFNXT(NR,NNAM,NAMES,X)
      IDF=1
    2 NN=IST(IDF)
      DO 4 I=1,IDF
      IF(MAT(NN).NE.NAMES(I)) GOTO 90
      IF(TYP(NN).NE.NAMES(I+IDF)) GOTO 90
    4 NN=NN+1
      NN=N1(1)
      IF(NN.EQ.0) GOTO 12
      DO 10 I=1,NN
      IF(FURNAM(I).NE.NAMES((I+1)*IDF+1))GOTO 90
   10 CONTINUE
   12 IF(NP.EQ.N1(4)) GOTO 30
      NR=1
      IF(IDZ.NE.IDENT) GOTO 40
   14 NN=N1(2)+N1(3)
      DO 20 I=1,NN
      LW=LW+1
      IF(LW.GT.LREC) CALL LDFREC
   20 X(I)=W(LW)
      NP=NP+1
      GOTO 100
C
C          END OF DATA FOR THIS NAMECOMBINATION.
C          IF FURTHER ENTRIES FOR SAME FIRST TWO NAMES EXIST,
C          LOAD POINTERS AND FIRST DATAPOINT OF NEXT.
   30 NR=0
      IF(N1(1).EQ.0) GOTO 100
      IF(N2(1).LE.NFN) GOTO 100
      NM=N2(2)*LREC+N2(3)+NFN*(N1(1)+3)
      I=(NM-1)/LREC
      LR=I
      CALL LDFREC
      LW=NM-I *LREC
      NN=N1(1)
      DO 32 I=1,NN
      IF(LW.GT.LREC) CALL LDFREC
      FURNAM(I)=W(LW)
      NAMES((I+1)*IDF+1)=FURNAM(I)
   32 LW=LW+1
      DO 34 I=1,3
      IF(LW.GT.LREC) CALL LDFREC
      N1(I+3)=IW(LW)
   34 LW=LW+1
      NFN=NFN+1
      LR=N1(5)
      CALL LDFREC
      LW=N1(6)-1
      NP=0
      GOTO 14
C
C          RELOAD COMMON/LDFRE/
```

```
   40 NM=N1(5)*LREC+N1(6)+NP*(N1(2)+N1(3))
      I=(NM-1)/LREC
      LR=I
      CALL LDFREC
      LW=NM-I*LREC-1
      GOTO 14
   90 NRETCD=16
      NR=0
      NN=IST(IDF)
      DO 91 I=1,IDF
      XM(NN)=NAMES(I)
      XT(NN)=NAMES(I+IDF)
   91 NN=NN+1
      IF(NN.NE.3) NN=1
      IDF=3-IDF
      DC 93 I=1,IDF
      XM(NN)=0
      XT(NN)=0
   93 NN=NN+1
      NN=4
      IF(NN.GT.NNAM(1)-2) NN=NNAM(1)-2
      INAM=NN
      IF(NN.EQ.0) GOTO 100
      DO 92 I=1,NN
   92 XF(I)=NAMES(I+2)
  100 RETURN
      END
```

```
      SUBROUTINE LDFCTB(IX,LT,IWT,IT)
      DIMENSION IWT(1),IT(1),IX(1)
      COMMON/LDFRE/ ID,IW(880),LW
      COMMON/LDFIL/IFIL,LRECD,LR
      REAL*8 ID
C
C           FETCH CONVERSION TABLE.
      LR=IX(1)
      READ(IFIL'LR)IW
      J=1
      LW=IX(2)-1
      DO 10 I=1,LT
      LW=LW+1
      IF(LW.GT.LRECD) CALL LDFREC
      IWT(J)=IW(LW)
      LW=LW+1
      IF(LW.GT.LRECD) CALL LDFREC
      IWT(J+1)=IW(LW)
      LW=LW+1
      IF(LW.GT.LRECD) CALL LDFREC
      IT(I)=IW(LW)
   10 J=J+2
      RETURN
      END
      SUBROUTINE LDFREC
      COMMON/LDFRE/ID,RW(440),LW
      REAL*8 RW,ID
      COMMON/LDFIL/IFIL,LREC,LR
      READ(IFIL'LR) RW
      LW=1
      RETURN
      END
```

```
      SUBROUTINE LDFMAT(IX,LX,IT,NT,AT)
      DIMENSION IX(1),IT(1),NT(1),AT(2,1)
      COMMON/LDFRC/NRETCD
      COMMON/LDFRE/ID,IW(880),LW
      COMMON/LDFIL/IFIL,LRECD,LR
      DATA IMAX/32767/
      INTEGER*2 NT,AT
      REAL*8 ID
C
C          FETCH MATERIAL-ADRESSTABLE.
      LR=IX(1)
      READ(IFIL'LR) IW
      LW=IX(2)-1
      DO 10 I=1,LX
      LW=LW+1
      IF(LW.GT.LRECD) CALL LDFREC
      IT(I)=IW(LW)
      LW=LW+1
      IF(LW.GT.LRECD) CALL LDFREC
      IF(IW(LW).GT.IMAX) GO TO 91
      NT(I)=IW(LW)
      DO 5 J=1,2
      LW=LW+1
      IF(LW.GT.LRECD) CALL LDFREC
      IF(IW(LW).GT.IMAX) GO TO 92
    5 AT(J,I)=IW(LW)
   10 CONTINUE
      NRETCD=0
      GOTO 100
   91 NRETCD=8
      GOTO 100
   92 NRETCD=9
      GOTO 100
  100 RETURN
      END
      SUBROUTINE LDFFNA(NAMES,CHNT,IDF)
      REAL NAMES(1),XM(3),XT(3),MAT(3),TYP(3)
      REAL*8 ID
      DIMENSION IFUR(5),FUR(5)
      EQUIVALENCE(FUR(1),IFUR(1))
      COMMON/LDFRE/ID,IW(880),LW
      COMMON/LDFIL/IFIL,LRECD,LR
      COMMON/LDFRC/NRETCD,IN,XM,XT,XF(4)
      COMMON/LDFRQ/MAT,TYP,FNAM(4),NAM(6),NFNAM(3),N1,N2
      LOGICAL CHNT
C
C           NEW NAME COMBINATION?
      NUMF=NAM(1)
      IF(.NOT.CHNT) GOTO 70
      DO 62 I=1,NUMF
      IF(FNAM(I).NE.NAMES((I+1)*IDF+1)) GOTO 71
   62 CONTINUE
      GOTO 99
C
C           FETCH POINTERS IN CASE OF MORE THAN TWO NAMES FROM THE FILE.
C           DO NOT STORE WHOLE TABLE.
   70 DO 73 I=1,3
   73 NFNAM(I)=NAM(I+3)
   71 DO 72 I=1,NUMF
   72 FNAM(I)=NAMES(I+2)
      LR=NFNAM(2)
      READ(IFIL'LR) IW
      LW=NFNAM(3)-1
      MAX=NFNAM(1)
C
C SEARCH FOR FIRST COMBINATION,WITH ALL NUMERIC NAMES.GE. GIVEN.
      DO 50 I=1,MAX
      DO 10 J=1,NUMF
      LW=LW+1
      IF(LW.GT.LRECD) CALL LDFREC
   10 IFUR(J)=IW(LW)
      DO 20 J=1,NUMF
      IF(FNAM(J).GT.FUR(J)) GO TO 40
   20 CONTINUE
C
C           FOUND.STORE NAMES AND POINTERS.
      DO 25 J=1,NUMF
      NAMES((J+1)*IDF+1)=FUR(J)
   25 FNAM(J)=FUR(J)
      N2=I
      DO 30 J=1,3
      LW=LW+1
      IF(LW.GT.LRECD) CALL LDFREC
   30 NAM(J+3)=IW(LW)
      GOTO 99
C
C           SKIP ENTRIES FOR POINTERS.
   40 DO 42 J=1,3
      LW=LW+1
      IF(LW.GT.LRECD) CALL LDFREC
   42 CONTINUE
   50 CONTINUE
C
C         NOT FOUND.STORE RETURNCODE AND ERROR-ANALYSIS INFORMATION.
      NRETCD=15
      DO 60 I=1,NUMF
      XF(I)=FNAM(I)
   60 FNAM(I)=0
      DO 63 I=1,3
      XT(I)=TYP(I)
   63 XM(I)=MAT(I)
      IN=NUMF
      GOTO 100
   99 NRETCD=0
  100 RETURN
      END




      SUBROUTINE LDFMAC(M1,I1,L1,M2,I2,L2)
      DIMENSION M1(1),I1(1),M2(1),I2(1)
      REAL*8 M1,M2,B/'        '/
C
C         CONVERT MATERIALADRESSTABLE TO ALPHAMERIC MATERIAL NAMES.
      DO 100 L=1,L2
      DO 10 J=1,L1
      IF(I2(L).NE.I1(J)) GOTO 10
      M2(L)=M1(J)
      GOTO 100
   10 CONTINUE
      M2(L)=B
  100 CONTINUE
      RETURN
      END




      SUBROUTINE LDFTB1(IDAT,IFL)
      COMMON/LDFTC/IDTYC(4),TYPNAM(70),ITYP(70),LTYCON
      COMMON/LDFMT/IDMAT(4),MATNAM(120),IMAT,NTYP(120),ATYP(2,120),
     *LMATAB
      DIMENSION IDENT(3),IDX(3),AMACON(2),ATYCON(2),AMATAB(2),IDFIL(5),
     1 MANAM(120),IMANAM(120),IMAT(120)
      INTEGER*2 NTYP,ATYP
      REAL*8 MATNAM,TYPNAM,MANAM
      INTEGER AMACON,AMATAB,ATYCON
      COMMON/LDFIL/IFIL
      COMMON/LDFRC/NRETCD
      DATA IDENT/'KEDA','BIBL','IDTH'/,IDFIL/'1111','2222','3333','4444'
     1          ,'5555'/
      DATA MMACON,MTYCON,MMATAB/120,70,120/
C
C         LOAD INFORMATION FROM MATERIAL-CONVERSIONTABLE,TYPE-CONVERSION
```

```
C           TABLE,MATERIAL-ADRESSTABLE INTO LDFTC,LDFMT.
C           USE MATERIAL-CONVERSIONTABLE TO CREATE ALPHA-MATERIAL-ADRESS
C           TABLE.
      NRETCD=0
      DO 10 I=1,3
      IF(IDTYC(I).NE.IDENT(I).OR.IDMAT(I).NE.IDENT(I)) GOTO 20
   10 CONTINUE
      IF(IDTYC(4).NE.IDFIL(IFL).OR.IDMAT(4).NE.IDFIL(IFL)) GOTO 20
C
C           TABLES ALREADY LOADED.
      GOTO 1000
C
C           TEST FILE IDENTIFICATION
   20 READ(IFIL'1) IDX,IDAT,LMACON,AMACON,LTYCON,ATYCON,LMATAB,AMATAB
      DO 22 I=1,3
      IF(IDENT(I).NE.IDX(I)) GOTO 901
   22 CONTINUE
C
C           TEST AVAILABLE STORAGE.
      IF(LMACON.GT.MMACON) GOTO 902
      IF(LTYCON.GT.MTYCON) GOTO 903
      IF(LMATAB.GT.MMATAB) GOTO 904
C
C           FETCH TABLES.
      CALL LDFCTB(AMACON,LMACON,MANAM,IMANAM)
      CALL LDFCTB(ATYCON,LTYCON,TYPNAM,ITYP)
      CALL LDFMAT(AMATAB,LMATAB,IMAT,NTYP,ATYP)
      IF(NRETCD.NE.0) GOTO 1000
      CALL LDFMAC(MANAM,IMANAM,LMACON,MATNAM,IMAT,LMATAB)
      DO 30 I=1,3
      IDTYC(I)=IDENT(I)
   30 IDMAT(I)=IDENT(I)
      IDTYC(4)=IDFIL(IFL)
      IDMAT(4)=IDFIL(IFL)
      GOTO 1000
  901 NRETCD=1
      GOTO 1000
  902 NRETCD=2
      GOTO 1000
  903 NRETCD=3
      GOTO 1000
  904 NRETCD=4
      GOTO 1000
 1000 RETURN
      END




      SUBROUTINE LDFTYC(IDF)
      COMMON/LDFTT/MAT,ITYNAM(70),ITYTAB(6,70),LTYTAB
      COMMON/LDFTC/DUM(4),TYPNAM(140),ITYP(70),LTYCON
      COMMON/LDFRC/NRETCD,IDUM,MX,TX
      COMMON/LDFRQ/MAT1,TYP,FNAM(4),NAM(6)
      INTEGER MAT(3),TX(3),MAT1(3),MX(3),TYP(3),TYPNAM,IST(2)/3,1/
      EQUIVALENCE(INAM,TYP(3))
      INTEGER*2 ITYTAB
C
C         FETCH POINTERS FOR'TYP'.
      IF(IDF.EQ.1) GOTO 15
      DO 10 I=1,LTYCON
      J=2*I-1
      IF(TYPNAM(J).NE.TYP(1)) GOTO 10
      IF(TYPNAM(J+1).NE.TYP(2)) GOTO 10
      INAM=ITYP(I)
      GOTO 20
   10 CONTINUE
   11 NRETCD=12
      GOTO 90
   15 DC 16 I=1,LTYCON
      IF(ITYP(I).NE.INAM) GOTO 16
      J=2*I-1
      TYP(1)=TYPNAM(J)
      TYP(2)=TYPNAM(J+1)
      GCTO 20
   16 CONTINUE
      GOTO 11
   20 DC 30 I=1,LTYTAB
      IF(ITYNAM(I).NE.INAM) GOTO 30
      NRETCD=0
      DC 25 J=1,6
   25 NAM(J)=ITYTAB(J,I)
      GCTO 100
   30 CONTINUE
      NRETCD=13
   90 J=IST(IDF)
      DO 92 I=1,IDF
      TX(J)=TYP(J)
      MX(J)=MAT(J)
   92 J=J+1
      IF(J.NE.3) J=1
      IDF=IDF-3
      DC 94 I=1,IDF
      TX(J)=0
      MX(J)=0
   94 J=J+1
      DO 96 I=1,3
   96 TYP(I)=0
  100 RETURN
      END
```

```
      SUBROUTINE LDFTAT(LTYP)
      COMMON/LDFTT/MAT,IMAT,ITYNAM(70),ITYTAB(6,70),LTYTAB
      COMMON/LDFRE/ID,IW(380),LW
      COMMON/LDFIL/IFIL,LREC
      COMMON/LDFRC/NRETCD
      REAL*8 MAT,ID
      INTEGER*2 ITYTAB
      DATA IMAX/32767/
C
C           FETCH TYPE-ADRESSTABLE FCR 'MAT'.
      I=LW
      CALL LDFREC
      LW=I
      LTYTAB=LTYP
      DC 10 I=1,LTYTAB
      LW=LW+1
      IF(LW.GT.LREC) CALL LDFREC
      ITYNAM(I)=IW(LW)
      DO 5 L=1,6
      LW=LW+1
      IF(LW.GT.LREC) CALL LDFREC
      IF(IW(LW).GT.IMAX) GO TO 91
    5 ITYTAB(L,I)=IW(LW)
   10 CONTINUF
      NRETCD=0
      RFTURN
   91 NRETCD=11
      RETURN
      END




      SUBROUTINE LDFTYP(IDF)
      COMMON/LDFTT/MAT,IMAT
      COMMON/LDFRE/ID,IW(880),LW
      COMMON/LDFIL/IFIL,LREC,LR
      CCMMON/LDFRC/NRETCD,IDUM,MX,IMX
      COMMON/LDFMT/DUN(4),MATNAM(120),IMANAM(120),NTYP(120),ATYP(2,120),
     *LMAT
     1AB
      INTEGER*2 NTYP,ATYP
      DATA MTYTAB/70/
      REAL*8 MAT,MX,MATNAM,ID
C
C           FETCH POINTERS TO TYPE-ADRESSTABLE FOR 'MAT'.
      DO 1 L=1,LMATAB
      IF(IDF.EQ.1) GOTO 7
      IF(MATNAM(L).NE.MAT) GOTO 1
      IMAT=IMANAM(L)
    6 LR=ATYP(1,L)
      LW=ATYP(2,L)-1
      LTYTAB=NTYP(L)
      GOTO 2
    7 IF(IMANAM(L).NE.IMAT) GOTO 1
      MAT=MATNAM(L)
      GOTO 6
    1 CCNTINUE
      NRETCD=7
      GOTO 92
    2 IF(LTYTAB.LE.MTYTAB) GOTO 3
```

```
      NRETCD=10
      GCTO 92
    3 CALL LDFTAT(LTYTAB)
      IF(NRETCD.EQ.0) GOTO 100
   92 MX=MAT
      IMX=IMAT
      IF(IDF.EQ.1) MX=0
      IF(IDF.EQ.2) IMX=0
      MAT=0
      IMAT=0
  100 RETURN
      END
```

6. Description of LDFPAC

LDFPAC is a subroutine package for retrieval of KEDAK-data like IDFPAC but for alphameric names only. It contains the sixteen subroutines described for IDFPAC,slightly differing only in the respect that they do not account for calls by numeric names. LDFPAC uses the same common blocks.

The main differences are: the absence of the IDF argument in the argument list of LDFNAM, LDFITN, LDFDPN, LDFTYP, LDFTYC, LDFFNA, and the absence of the IDF-ENTRY in LDFLOC and LDFNXT.

The subroutine names with the argument lists:

```
1. LDFOPN (IFI, IDAT*)
   ENTRIES: LDFALF (IFS, IFI,*)
            LDFROP (IFI,*)

2. LDFLOC (NR, NNAM, NAMES,*)
   ENTRIES: LDFTTO (NR, NAMES)
            LDFRQO (NR, NAMES)

3. LDFNXT (NR, NNAM, NAMES,*)

4. LDFITN (NR, MAT, TYPES, NUM, MAX, MMAX)

5. LDFNAM (NR, NAMES, XNAM, N1, N2, MAX)

6. LDFERR

7. LDFDPN (NR, NAMES, NUM)

8. LDFTB1 (IDAT, IFL)

9. LDFCTB (IX, LT, IWT, IT)
```

10. LDFMAC (M1, I1, L1, M2, I2, L2)

11. LDFMAT (IX, LX, IT, NT, AT)

12. LDFTAT (LTYP)

13. LDFTYP

14. LDFFNA (NAMES, CHNT)

15. LDFTYC

16. LDFREC

The argument description given in the previous sections for IDFPAC do hold for these entry points in the same manner and the user is referred there for details.

The common blocks used in LDFPAC are generally organized in the same manner as the common blocks described for IDFPAC, the only exception being the common LDFMT where the array for numeric material names is missing.

Also the overlay structure and call pattern for LDFPAC is the same as for IDFPAC.

7. Source lists of LDFPAC

Subroutine name	Page
LDFERR	III-91
LDFLOC	III-92
LDFNXT	III-94
LDFREC	III-96
LDFMAT	III-96
LDFFNA	III-97
LDFITN	III-98
LDFDPN	III-99
LDFTYC	III-99
LDFMAC	III-100
LDFTB1	III-100
LDFNAM	III-101
LDFOPN	III-102
LDFTAT	III-103
LDFTYP	III-104
LDFCTB	III-105

```
      SUBROUTINE LDFERR
      COMMON/INOUT/KOUT
      COMMON/LDFRC/NRETCD,NXF,MAT,TYP,XF(4)
      COMMON/LDFIL/IFIL
      REAL*8 MAT,TYP
C
C           PRINT ERRORDIAGNOSTIC.
C         KOUT....PRINTING UNIT.
      NERR=0
      WRITE(KOUT,600)
  600 FORMAT(/T5,'ERRORDIAGNOSTIC FOR KEDAKRETRIEVAL:'/)
      IF(NRETCD.EQ.0) GOTO 910
      GOTO (1,2,3,4,5,6,7,8,9,10,11,12,13,14,15,16,17),NRETCD
      GOTO 900
    1 WRITE(KOUT,601) IFIL
  601 FORMAT(T10,'DATASET ON UNIT:',I2,' DOES NOT CONTAIN CORRECT KEDAK-
     1IDENTIFICATION'/)
      GOTO 1000
    2 WRITE(KOUT,602) IFIL
  602 FORMAT(T10,'LENGTH OF MATERIAL CONVERSION TABLE ON UNIT:',I2,' EXC
     1EEDS AVAILABLE STORAGE.CALL PROGRAMMER.'/)
      GOTO 1000
    3 WRITE(KOUT,603) IFIL
  603 FORMAT(T10,'LENGTH OF TYPE CONVERSION TABLE ON UNIT:',I2,' EXCEEDS
     1 AVAILABLE STORAGE.CALL PROGRAMMER.'/)
      GOTO 1000
    4 WRITE(KOUT,604) IFIL
  604 FORMAT(T10,'LENGTH OF MATERIAL ADRESS TABLE ON UNIT:',I2,' EXCEEDS
     1 AVAILABLE STORAGE.CALL PROGRAMMER.'/)
      GOTO 1000
    5 WRITE(KOUT,605)
  605 FORMAT(T10,'FILE NUMBER USED IN LDFOPN IS OUT OF VALID RANGE.'/)
      GOTO 1000
    6 WRITE(KOUT,606)
  606 FORMAT(T10,'LDFLOC CALL WAS EXECUTED,BUT NO FILE WAS OPENED BEFORE
     1.'/)
      GOTO 1000
    7 WRITE(KOUT,607)MAT
  607 FORMAT(T10,'REQUESTED MATERIAL:',A8,' NOT ON AVAILABLE FILES.'/)
      GOTO 1000
    8 WRITE(KOUT,608) IFIL
  608 FORMAT(T10,'FIXPOINT OVERFLOW IN LENGTH-ITEM OF MATERIAL-ADRESSTAB
     1LE ON UNIT:',I2,'.CALL PROGRAMMER.'/)
      GOTO 1000
    9 WRITE(KOUT,609) IFIL
  609 FORMAT(T10,'FIXPOINT OVERFLOW IN ADRESS-ITEM OF MATERIAL-ADRESSTAB
     1LE ON UNIT:',I2,'.CALL PROGRAMMER.'/)
      GOTO 1000
   10 WRITE(KOUT,610) MAT
  610 FORMAT( T10,'LENGTH OF TYPE-ADRESSTABLE FOR ',A8,' EXCEEDS AVAILAB
     1LE STORAGE.CALL PROGRAMMER.'/)
      GOTO 1000
   11 WRITE(KOUT,611) MAT
  611 FORMAT( T10,'FIXPOINT OVERFLOW IN WORDS 2-7 OF TYPE-ADRESS TABLE F
     1OR ',A8,'.CALL PROGRAMMER.'/)
```

```
      GOTO 1000
   12 WRITE(KOUT,612) IFIL,TYP
  612 FORMAT( T10,'TYPE-CONVERSIONTABLE ON UNIT:',I2,' DOES NOT CONTAIN
     1',A8,'.'/)
      GOTO 1000
   13 WRITE(KOUT,613) MAT,IFIL,TYP
  613 FORMAT(T10,'MATERIAL ',A8,' ON UNIT ',I2,' DOES NOT CONTAIN TYPE '
     1,A8,'.'/)
      GOTO 1000
   14 WRITE(KOUT,614) MAT,TYP,IFIL
  614 FORMAT(T10,'NUMBER OF FURTHER NAMES FOR MATERIAL:',A8,',TYPE:',A8,
     1',ON UNIT ',I2,' EXCEEDS AVAILABLE STORAGE.'//
     2T25,'CALL PROGRAMMER.'/)
      GOTO 1000
   15 WRITE(KOUT,615) MAT,TYP,IFIL,(XF(I),I=1,NXF)
  615 FORMAT(T10,'FOR MATERIAL:',A8,',TYPE:',A8,', ON UNIT ',I2,' AT LE
     1AST ONE OF THE REQUESTED NUMERICAL NAMES'//
     2T10,'IS ABOVE THE LARGEST ONE STORED ON KEDAK.'//
     3T10,'THE NUMERICAL NAMES REQUESTED ARE :',1P4E13.5/)
      GOTO 1000
   16 WRITE(KOUT,616) MAT,TYP
      IF(NXF.NE.0) WRITE(KOUT,902) (XF(I),I=1,NXF)
  616 FORMAT(T10,'LDFNXT WAS CALLED WITHOUT PREVIOUS SUCCESSFUL EXECUTIO
     1N OF LDFLOC FOR:'//
     2 T20,'MATERIAL:',A8,'     TYPE:',A8/)
  902 FORMAT(T20,'NUMERIC NAMES (IF ANY):',1P4E13.5/)
      GOTO 1000
   17 WRITE(KOUT,617)
  617 FORMAT(T10,'ATTEMPT TO OPEN A NEW FILE BY CALL TO LDFOPN,THOUGH MA
     1XIMUM NUMBER HAS ALREADY BEEN OPENED.'/)
      GOTO 1000
  900 WRITE(KOUT,901) NRETCD
  901 FORMAT(T10,'NO MESSAGE STORED FOR THIS ERRORTYPE.RETURNCODE WAS:',
     1I3/)
      GOTO 1000
  910 WRITE(KOUT,911)
  911 FORMAT(T10,'NO ERRORCONDITION OCURRED.'/)
 1000 RETURN
      END




      SUBROUTINE LDFLOC(NR,NNAM,NAMES,X)
      DIMENSION NNAM(1),NAMES(1),X(1)
      REAL*8 NAMES,MAT,XM,TYP,XT,MAT1,XNAM,IDZ,IDENT/'KEDABIBL'/
      INTEGER X
      INTEGER*2 ITYTAB
      LOGICAL TTONLY,NODATA,CHNT
      COMMON/LDFTT/MAT,ITYNAM(70),ITYTAB(6,70),LTYTAB
      COMMON/LDFRC/NRETCD,IDUM,XM,XT,XF(4)
      COMMON/LDFRE/IDZ,IW(880),LW
      COMMON/LDFIL/IFIL,LRECD,LR
      COMMON/LDFRQ/MAT1,TYP,FNAM(4),NAM,NAMX(8),NRDONE,NFDONE
      DATA MFNAM/4/
```

```
C
C          LDFLOC: LOCATE DATA OF REQUESTED NAMES.
C                  PROVIDE FIRST DATA POINT.
      TTONLY=.FALSE.
      NODATA=.FALSE.
      GOTO 10
C
C          PROVIDE TYPE-ADRESSTABLE ONLY.
      ENTRY LDFTTO(NR,NAMES)
      TTONLY=.TRUE.
      GOTO 10
C
C          PROVIDE CONTENTS OF LDFRQ ONLY.
      ENTRY LDFRQO(NR,NAMES)
      TTONLY=.FALSE.
      NODATA=.TRUE.
      GOTO 10
C
C          REDO LDFOPN,IF NECESSARY.
   10 CALL LDFROP(IFIL,&999)
C
C          LOAD TYPE-ADRESSTABLE IF NECESSARY.
      IF(MAT.EQ.NAMES(1).AND.MAT1.EQ.NAMES(1)) GOTO 30
      IF(MAT.EQ.NAMES(1)) GOTO 32
      I=IFIL
   12 MAT=NAMES(1)
      CALL LDFTYP
      IF(NRETCD.EQ.7) GOTO 20
      IF(NRETCD.NE.0) GOTO 999
      GOTO 32
C
C          MATERIAL NOT ON CURRENT UNIT.TRY ALTERNATIVE UNIT.
   20 CALL LDFALF(I,IFIL,&999)
      GOTO 12
C
C          FETCH POINTERS TO REQUESTED TYPE,IF NECESSARY.
   30 IF(TTONLY)  GOTO 1001
      IF(TYP.EQ.NAMES(2)) GOTO 40
      GOTO 33
   32 IF(TTONLY) GOTO 1001
   33 TYP=NAMES(2)
      MAT1=NAMES(1)
      CALL LDFTYC
      IF(NRETCD.NE.0) GOTO 999
      IF(NAM.EQ.0) GOTO 90
      IF(NAM.GT.MFNAM) GOTO 42
      CHNT=.FALSE.
      GOTO 60
C
C          FURTHER NAMES?
   40 IF(NAM.EQ.0) GOTO 90
      IF(NAM.GT.MFNAM) GOTO 42
      CHNT=.TRUE.
      GOTO 60
   42 NRETCD=14
```

```
      XM=MAT
      XT=TYP
      GOTO 999
   60 CALL LDFFNA(NAMES,CHNT)
      IF(NRETCD.NE.0) GOTO 999
C
C          FILL NNAM.
   90 NRDONE=0
      IF(NODATA) GOTO 1001
      NNAM(1)=NAM+2
      NNAM(2)=NAMX(1)
      NNAM(3)=NAMX(2)
C
C          FETCH FIRST RECORD.
      LR=NAMX(4)
      LW=NAMX(5)-1
      READ(IFIL'LR) IW
      IDZ=IDENT
      NUMX=NAMX(1)+NAMX(2)
      DO 92 I=1,NUMX
      LW=LW+1
      IF(LW.GT.LRECD) CALL  LDFREC
   92 X(I)=IW(LW)
      NRDONE=1
 1001 NR=1
      GOTO 1000
  999 NR=0
 1000 RETURN
      END




      SUBROUTINE LDFNXT(NR,NNAM,NAMES,X)
      DIMENSION NAMES(1),X(1),NNAM(1)
      COMMON/LDFRE/IDZ,W(880),LW
      COMMON/LDFRC/NRETCD,INAM,XM,XT,XF(4)
      COMMON/LDFRQ/MAT,TYP,FURNAM(4),N1(6),N2(3),NP,NFN
      COMMON/LDFIL/IFIL,LREC,LR
      REAL*8 MAT,TYP,NAMES,XM,XT,FN,IDZ,IDENT/'KEDABIBL'/
      INTEGER IW(880)
      EQUIVALENCE (IW(1),W(1))
C
C          RETRIEVE NEXT DATAPOINT.
      NN=N1(1)
      IF(MAT.NE.NAMES(1)) GOTO 90
      IF(TYP.NE.NAMES(2)) GOTO 90
      IF(NN.EQ.0) GOTO 12
      DO 10 I=1,NN
      FN=FURNAM(I)
      IF(FN.NE.NAMES(I+2)) GOTO 90
   10 CONTINUE
   12 IF(NP.EQ.N1(4)) GOTO 30
      NR=1
      IF(IDZ.NE.IDENT) GOTO 40
   14 NN=N1(2)+N1(3)
      DC 20 I=1,NN
      LW=LW+1
      IF(LW.GT.LREC) CALL LDFREC
   20 X(I)=W(LW)
      NP=NP+1
      GCTO 100
C
C          END OF DATA FOR THIS NAMECOMBINATION.
C          IF FURTHER ENTRIES FOR SAME FIRST TWO NAMES EXIST,
C          LOAD POINTERS AND FIRST DATAPOINT OF NEXT.
   30 NR=0
      IF(N1(1).EQ.0) GOTO 100
      IF(N2(1).LE.NFN) GOTO 100
      NM=N2(2)*LREC+N2(3)+NFN*(N1(1)+3)
      I=(NM-1)/LREC
      LR=I
      CALL LDFREC
      LW=NM-I*LREC
      NN=N1(1)
      DO 32 I=1,NN
      IF(LW.GT.LREC) CALL LDFREC
      FURNAM(I)=W(LW)
      NAMES(I+2)=FURNAM(I)
   32 LW=LW+1
      DC 34 I=1,3
      IF(LW.GT.LREC) CALL LDFREC
      N1(I+3)=IW(LW)
   34 LW=LW+1
      NFN=NFN+1
      LR=N1(5)
      CALL LDFREC
      LW=N1(6)-1
      NP=0
      GOTO 14
C
C          RELOAD COMMON/LDFRE/
   40 NM=N1(5)*LREC+N1(6)+NP*(N1(2)+N1(3))
      I=(NM-1)/LREC
      LR=I
      CALL LDFREC
      LW=NM-I *LREC-1
      GCTO 14
   90 NRETCD=16
      NR=0
      NN=4
      IF(NN.GT.NNAM(1)-2) NN=NNAM(1)-2
      XM=NAMES(1)
      XT=NAMES(2)
      INAM=NN
      IF(NN.EQ.0) GOTO 100
      DO 92 I=1,NN
   92 XF(I)=NAMES(I+2)
  100 RETURN
      END
      SUBROUTINE LDFREC
      COMMON/LDFRE/ID,RW(440),LW
      REAL*8 RW,ID
      COMMON/LDFIL/IFIL,LREC,LR
      READ(IFIL'LR) RW
      LW=1
      RETURN
      END
```

```
      SUBROUTINE LDFMAT(IX,LX,IT,NT,AT)
      DIMENSION IX(1),IT(1),NT(1),AT(2,1)
      COMMON/LDFRC/NRETCD
      COMMON/LDFRE/ID,IW(880),LW
      COMMON/LDFIL/IFIL,LRECD,LR
      DATA IMAX/32767/
      INTEGER*2 NT,AT
      REAL*8 ID
C
C         FETCH MATERIAL-ADRESSTABLE.
      LR=IX(1)
      READ(IFIL'LR) IW
      LW=IX(2)-1
      DO 10 I=1,LX
      LW=LW+1
      IF(LW.GT.LRECD) CALL LDFREC
      IT(I)=IW(LW)
      LW=LW+1
      IF(LW.GT.LRECD) CALL LDFREC
      IF(IW(LW).GT.IMAX) GO TO 91
      NT(I)=IW(LW)
      DO 5 J=1,2
      LW=LW+1
      IF(LW.GT.LRECD) CALL LDFREC
      IF(IW(LW).GT.IMAX) GO TO 92
    5 AT(J,I)=IW(LW)
   10 CONTINUE
      NRETCD=0
      GOTO 100
   91 NRETCD=8
      GOTO 100
   92 NRETCD=9
      GOTO 100
  100 RETURN
      END
      SUBROUTINE LDFFNA(NAMES,CHNT)
      DIMENSION IFUR(5),FUR(5)
      EQUIVALENCE(FUR(1),IFUR(1))
      COMMON/LDFRE/ID,IW(880),LW
      COMMON/LDFIL/IFIL,LRECD,LR
      COMMON/LDFRC/NRETCD,IN,XM,XT,XF(4)
      COMMON/LDFRQ/MAT,TYP,FNAM(4),NAM(6),NFNAM(3),N1,N2
      REAL*8 TYP,MAT,XT,XM,NAMES(1),XNAM,ID
      LOGICAL CHNT
      DATA F0/'0000'/
C
C          NEW NAME COMBINATION?
      NUMF=NAM(1)
      IF(.NOT.CHNT) GOTO 70
      DO 62 I=1,NUMF
      XNAM=FNAM(I)
      IF(XNAM.NE.NAMES(I+2)) GOTO 71
   62 CONTINUE
      GOTO 99
C
C          FETCH POINTERS IN CASE OF MORE THAN TWO NAMES FROM THE FILE.
C          DO NOT STORE WHOLE TABLE.
   70 DO 73 I=1,3
   73 NFNAM(I)=NAM(I+3)
   71 DO 72 I=1,NUMF
   72 FNAM(I)=NAMES(I+2)
      LR=NFNAM(2)
      READ(IFIL'LR) IW
      LW=NFNAM(3)-1
      MAX=NFNAM(1)
C
C SEARCH FOR FIRST COMBINATION,WITH ALL NUMERIC NAMES.GE. GIVEN.
      DO 50 I=1,MAX
      DO 10 J=1,NUMF
      LW=LW+1
      IF(LW.GT.LRECD) CALL LDFREC
   10 IFUR(J)=IW(LW)
      DO 20 J=1,NUMF
      IF(FNAM(J).GT.FUR(J)) GO TO 40
   20 CONTINUE
C
C          FOUND.STORE NAMES AND POINTERS.
      DO 25 J=1,NUMF
      NAMES(2+J)=FUR(J)
   25 FNAM(J)=FUR(J)
      N2=I
      DO 30 J=1,3
      LW=LW+1
      IF(LW.GT.LRECD) CALL LDFREC
   30 NAM(J+3)=IW(LW)
      GOTO 99
C
C          SKIP ENTRIES FOR POINTERS.
   40 DO 42 J=1,3
      LW=LW+1
      IF(LW.GT.LRECD) CALL LDFREC
   42 CONTINUE
   50 CONTINUE
C
C         NOT FOUND.STORE RETURNCODE AND ERROR-ANALYSIS INFORMATION.
      NRETCD=15
      DO 60 I=1,NUMF
      XF(I)=FNAM(I)
   60 FNAM(I)=F0
      XT=TYP
      XM=MAT
      IN=NUMF
      GOTO 100
   99 NRETCD=0
  100 RETURN
      END
```

```
      SUBROUTINE LDFITN(NR,MAT,TYPES,NUM,MAX,MMAX)
      REAL*8 MAT,TYPES(1),M0,TYPNAM
      DIMENSION NUM(1)
      COMMON/LDFTT/M0,ITYNAM(70),ITYTAB(6,70),LTYTAB
      INTEGER*2 ITYTAB
      COMMON/LDFTC/DUM(4),TYPNAM(70),ITYP(70),LTYCON
C
C         PROVIDE SOME INFORMATION FROM TYPE-ADRESSTABLE.
C        MAT.....MATERIAL NAME.
C        TYPES.....AVAILABLE TYPE-NAMES FOR THAT MATERIAL.
C        NUM(I).....NUMBER OF DATAPOINTS FOR I-TH TYPE.IF THIS TYPE
C              HAS FURTHER NAMES,NUM(I) REFLECTS THE NUMBER OF NAME
C              COMBINATIONS FOR THAT TYPE,E.G. FOR SGNC THE NUMBER
C              OF DISTRIBUTIONS.
C        MAX.....NUMBER OF AVAILABLE TYPES.
C        MMAX.....LENGTH OF ARRAYS TYPES(R*8),NUM(I*4).
C        NR.....RETURN-CODE: 0=ERROR. 1=NO ERROR.
C              IF NR=0,MAX>0 NUMBER OF TYPES EXCEEDS LENGTH OF ARRAYS.
C
      CALL LDFTTO(NR,MAT)
      IF(NR.EQ.0) GOTO 110
      MAX=LTYTAB
      IF(MAX.GT.MMAX) MAX=MMAX
      DO 100 L=1,LTYTAB
      IF(L.GT.MMAX) GOTO 120
      DO 10 I=1,LTYCON
      IF(ITYP(I).NE.ITYNAM(L)) GOTO 10
      TYPES(L)=TYPNAM(I)
      NUM(L)=ITYTAB(4,L)
      GOTO 100
   10 CONTINUE
      TYPES(L)=BL
      NUM(L)=0
  100 CONTINUE
      NR=1
      GOTO 1000
  110 MAX=0
      GOTO 1000
  120 NR=0
 1000 RETURN
      END




      SUBROUTINE LDFDPN(NR,NAMES,NUM)
      REAL*8 NAMES(1)
      INTEGER*2 ITYTAB
      DIMENSION NNAM(3)
      COMMON/LDFRQ/XDUM(11),NDP
C
C          PROVIDE NUMBER OF DATAPOINTS FOR GIVEN NAMECOMBINATION.
C        NAMES.....NAME COMBINATION.
C        NUM.....NUMBER OF DATAPOINTS.
C        NR.....RETURNCODE: 0=ERROR, 1=NO ERROR.
      NUM=0
      CALL LDFRQO(NR,NAMES)
      IF(NR.EQ.0) GOTO 10
      NUM=NDP
      NR=1
   10 RETURN
      END




      SUBROUTINE LDFTYC
      COMMON/LDFTT/MAT,ITYNAM(70),ITYTAB(6,70),LTYTAB
      COMMON/LDFTC/DUM(4),TYPNAM(70),ITYP(70),LTYCON
      COMMON/LDFRC/NRETCD,IDUM,MX,TX
      COMMON/LDFRQ/MAT1,TYP,FNAM(4),NAM(6)
      REAL*8 MX,TX,TYPNAM,MAT,TYP,MO/'00000000'/,MAT1
      INTEGER*2 ITYTAB
C
C          FETCH POINTERS FOR'TYP'.
      DO 10 I=1,LTYCON
      IF(TYPNAM(I).NE.TYP) GOTO 10
      INAM=ITYP(I)
      GOTO 20
   10 CONTINUE
      NRETCD=12
      GOTO 90
   20 DO 30 I=1,LTYTAB
      IF(ITYNAM(I).NE.INAM) GOTO 30
      NRETCD=0
      DO 25 J=1,6
   25 NAM(J)=ITYTAB(J,I)
      GOTO 100
   30 CONTINUE
      NRETCD=13
   90 TX=TYP
      MX=MAT
      TYP=MO
  100 RETURN
      END
```

```
      SUBROUTINE LDFMAC(M1,I1,L1,M2,I2,L2)
      DIMENSION M1(1),I1(1),M2(1),I2(1)
      REAL*8 M1,M2,B/'        '/
C
C          CONVERT MATERIALADRESSTABLE TO ALPHAMERIC MATERIAL NAMES.
      DO 100 L=1,L2
      DO 10 J=1,L1
      IF(I2(L).NE.I1(J)) GOTO 10
      M2(L)=M1(J)
      GOTO 100
   10 CONTINUE
      M2(L)=B
  100 CONTINUE
      RETURN
      END
```

```
      SUBROUTINE LDFTB1(IDAT,IFL)
      COMMON/LDFTC/IDTYC(4),TYPNAM(70),ITYP(70),LTYCON
      COMMON/LDFMT/IDMAT(4),MATNAM(120),NTYP(120),ATYP(2,120),LMATAB
      DIMENSION IDENT(3),IDX(3),AMACON(2),ATYCON(2),AMATAB(2),IDFIL(5),
     1 MANAM(120),IMANAM(120),IMAT(120)
      INTEGER*2 NTYP,ATYP
      REAL*8 MATNAM,TYPNAM,MANAM,MAT,TYP,MAT1
      INTEGER AMACON,AMATAB,ATYCON
      COMMON/LDFIL/IFIL
      COMMON/LDFRC/NRETCD
      COMMON/LDFTT/ MAT
      COMMON/LDFRQ/ MAT1,TYP
      DATA IDENT/'KEDA','BIBL','IOTH'/,IDFIL/'1111','2222','3333','4444'
     1,'5555'/
      DATA MMACON,MTYCON,MMATAB/120,70,120/
C
C          LOAD INFORMATION FROM MATERIAL-CONVERSIONTABLE,TYPE-CONVERSION
C          TABLE,MATERIAL-ADRESSTABLE INTO LDFTC,LDFMT.
C          USE MATERIAL-CONVERSIONTABLE TO CREATE ALPHA-MATERIAL-ADRESS
C          TABLE.
      NRETCD=0
      DO 10 I=1,3
      IF(IDTYC(I).NE.IDENT(I).OR.IDMAT(I).NE.IDENT(I)) GOTO 20
   10 CONTINUE
      IF(IDTYC(4).NE.IDFIL(IFL).OR.IDMAT(4).NE.IDFIL(IFL)) GOTO 20
C
C          TABLES ALREADY LOADED.
```

```
      GOTO 1000
C
C           TEST FILE IDENTIFICATION
   20 READ(IFIL'1) IDX,IDAT,LMACON,AMACON,LTYCON,ATYCON,LMATAB,AMATAB
      DO 22 I=1,3
      IF(IDENT(I).NE.IDX(I)) GOTO 901
   22 CONTINUE
C
C           TEST AVAILABLE STORAGE.
      IF(LMACON.GT.MMACON) GOTO 902
      IF(LTYCON.GT.MTYCON) GOTO 903
      IF(LMATAB.GT.MMATAB) GOTO 904
C     PURGE MAT,MAT1,TYP
C
      MAT=0.0D+0
      MAT1=0.0D+0
      TYP=0.0D+0
C
C           FETCH TABLES.
      CALL LDFCTB(AMACON,LMACON,MANAM,IMANAM)
      CALL LDFCTB(ATYCON,LTYCON,TYPNAM,ITYP)
      CALL LDFMAT(AMATAB,LMATAB,IMAT,NTYP,ATYP)
      IF(NRETCD.NE.0) GOTO 1000
      CALL LDFMAC(MANAM,IMANAM,LMACON,MATNAM,IMAT,LMATAB)
      DO 30 I=1,3
      IDTYC(I)=IDENT(I)
   30 IDMAT(I)=IDENT(I)
      IDTYC(4)=IDFIL(IFL)
      IDMAT(4)=IDFIL(IFL)
      GOTO 1000
  901 NRETCD=1
      GOTO 1000
  902 NRETCD=2
      GOTO 1000
  903 NRETCD=3
      GOTO 1000
  904 NRETCD=4
      GOTO 1000
 1000 RETURN
      END




      SUBROUTINE LDFNAM(NR,NAMES,XNAM,N1,N2,MAX)
      REAL*8 NAMES(1),MAT,TYP,ID
      DIMENSION XNAM(1)
      COMMON/LDFRQ/MAT,TYP,FNAM(4),NPD(6)
      COMMON/LDFRC/NRC
      COMMON/LDFIL/IFIL,LREC,LR
      COMMON/LDFRE/ ID,W(880),LW
C
C PROVIDE INFORMATION FROM ADRESSTABLE OF FURTHER NAMES.
C XNAM:FURTHER NAMES,N1:NUMBER/COMBINATION,N2:NUMBER OF COMBINATIONS.
C NAMES.....NAME COMBINATION FOR WHICH FURTHER NAMES ARE INQUIRED.
```

```
C NR.....RETURN CODE: 0=ERROR, 1=NO ERROR. NR=0,N2>0 INDICATES XNAM
C               TOO SHORT.
      N1=0
      N2=0
      CALL LDFTTO(NR,NAMES)
      IF(NR.EQ.0) GOTO 100
      NR=0
      MAT=NAMES(1)
      TYP=NAMES(2)
      CALL LDFTYC
      IF(NRC.NE.0) GOTO 100
      N1=NPO(1)
      IF(N1.EQ.0) GOTO 100
      LR=NPO(5)
      READ(IFIL'LR) W
      LW=NPO(6)-1
      NX=NPO(4)
      DO 10 I=1,NX
      DO 20 J=1,N1
      N2=N2+1
      IF(N2.GT.MAX) GOTO 90
      LW=LW+1
      IF(LW.GT.LREC) CALL LDFREC
   20 XNAM(N2)=W(LW)
      DO 30 J=1,3
      LW=LW+1
      IF(LW.GT.LREC) CALL LDFREC
   30 CONTINUE
   10 CONTINUE
      NR=1
      GOTO 92
   90 N2=N2-1
   92 N2=N2/N1
  100 TYP=0
      RETURN
      END




      SUBROUTINE LDFOPN(IFI,IDAT,*)
      COMMON/LDFRC/NRETCD
      COMMON/LDFIL/IFIL,LRECD,LR
      DIMENSION NFIL(5)
      DATA IFL/0/,LREC/880/,MAXFIL/5/,NFIL/5*0/,LDS/8000/
      DATA XU/'U'/
C
C  LDFOPN:DEFINES THE FILE,PROVIDES FIRST THREE TABLES AND KEEPS
C          SOME INFORMATION.IT MUST NOT BE OVERLAYED.
      NRETCD=0
      IF(IFI.LE.0.OR.IFI.GT.99) GOTO 90
      DO 2 I=1,MAXFIL
      IF(NFIL(I).EQ.IFI) GOTO 1
      IF(NFIL(I).NE.0) GOTO 2
      IFL=I
      NFIL(I)=IFI
      GOTO 4
    1 IFL=I
      GOTO 20
    2 CONTINUE
      NRETCD=17
      GOTO 100
    4 CALL DEFI(IFI,LDS,XU,LREC,LR)
      IF(LDS.GE.0) GOTO 20
      LDS=-LDS
      GOTO 90
C
C          REREAD TABLES OF CURRENT FILE.
      ENTRY LDFROP(IFI,*)
      IF(IFL.NE.0) GOTO 15
   12 NRETCD=6
      GOTO 100
   15 IFI=NFIL(IFL)
   20 IFIL=IFI
      LRECD=LREC
      CALL LDFTB1(IDAT,IFL)
      IF(NRETCD.NE.0) GOTO 100
   50 RETURN
   90 NRETCD=5
  100 RETURN 1
C
C          LOAD TABLES OF AN ALTERNATIVE FILE.
      ENTRY LDFALF(IFS,IFI,*)
      IF(IFL.EQ.0) GOTO 12
      IFLK=IFL
      IFL=IFL+1
      IF(IFL.GT.MAXFIL) IFL=1
      IF(NFIL(IFL).EQ.0) IFL=1
      IFI=NFIL(IFL)
      IF(IFI.NE.IFS) GOTO 20
      IFL=IFLK
      NRETCD=7
      GOTO 100
      END




      SUBROUTINE LDFTAT(LTYP)
      COMMON/LDFRE/ID,IW(880),LW
      COMMON/LDFTT/MAT,ITYNAM(70),ITYTAB(6,70),LTYTAB
      COMMON/LDFIL/IFIL,LREC
      COMMON/LDFRC/NRETCD
      REAL*8 ID,MAT
      INTEGER*2 ITYTAB
      DATA IMAX/32767/
C
C          FETCH TYPE-ADRESSTABLE FOR 'MAT'.
      I=LW
      CALL LDFREC
      LW=I
      LTYTAB=LTYP
      DO 10 I=1,LTYTAB
      LW=LW+1
      IF(LW.GT.LREC) CALL LDFREC
      ITYNAM(I)=IW(LW)
      DO 5 J=1,6
      LW=LW+1
      IF(LW.GT.LREC) CALL LDFREC
      IF(IW(LW).GT.IMAX) GO TO 91
    5 ITYTAB(J,I)=IW(LW)
   10 CONTINUE
      NRETCD=0
      RETURN
   91 NRETCD=11
      RETURN
      END
```

```
      SUBROUTINE LDFTYP
      COMMON/LDFTT/MAT
      COMMON/LDFRE/ID,IW(880),LW
      COMMON/LDFIL/IFIL,LREC,LR
      COMMON/LDFRC/NRETCD,IDUM,MX
      COMMON/LDFMT/DUN(4),MATNAM(120),NTYP(120),ATYP(2,120),LMATAB
      INTEGER*2 NTYP,ATYP
      DATA MTYTAB/70/
      REAL*8 MAT,MO/'00000000'/,MX,MATNAM,ID
C
C           FETCH POINTERS TO TYPE-ADRESSTABLE FOR 'MAT'.
      DO 1 L=1,LMATAB
      IF(MATNAM(L).NE.MAT) GOTO 1
      LR=ATYP(1,L)
      LW=ATYP(2,L)-1
      LTYTAB=NTYP(L)
      GOTO 2
    1 CONTINUE
      NRETCD=7
      GOTO 92
    2 IF(LTYTAB.LE.MTYTAB) GOTO 3
      NRETCD=10
      GOTO 92
    3 CALL LDFTAT(LTYTAB)
      IF(NRETCD.EQ.0) GOTO 100
   92 MX=MAT
      MAT=MO
  100 RETURN
      END
      SUBROUTINE LDFCTB(IX,LT,IWT,IT)
      DIMENSION IWT(1),IT(1),IX(1)
      COMMON/LDFRE/ID,IW(880),LW
      COMMON/LDFIL/IFIL,LRECD,LR
      REAL*8 ID
C
C           FETCH CONVERSION TABLE.
      LR=IX(1)
      READ(IFIL'LR)IW
      J=1
      LW=IX(2)-1
      DO 10 I=1,LT
      LW=LW+1
      IF(LW.GT.LRECD) CALL LDFREC
      IWT(J)=IW(LW)
      LW=LW+1
      IF(LW.GT.LRECD) CALL LDFREC
      IWT(J+1)=IW(LW)
      LW=LW+1
      IF(LW.GT.LRECD) CALL LDFREC
      IT(I)=IW(LW)
   10 J=J+2
      RETURN
      END
```

Appendix 1:

Material and type name conversion table, the table of available isotopes and the types available for each isotope listed by the program to print the contents of a KEDAK-file (see 2.2.3).

Date of printout: January, 1977.

```
                 THE TABLE OF CONTENTS OF THE KEDAK FILE

          FILE IDENTIFICATION :KEDABIBLIOTH     DATE :    201276



        THE MATERIAL CONVERSION TABLE CONTAINS  66 MATERIAL NAMES



ALPHAMERIC NUMERIC  ALPHAMERIC NUMERIC  ALPHAMERIC NUMERIC

H    1       10001  H   H1       11001  H   D1       12001
H    2       10002  HE   3       20003  HE   4       20004
C   12       60012  N            70000  O   16       80016
NA 23       110023  AL  27      130027  CR          240000
CR 50       240050  CR  52      240052  CR  53      240053
CR 54       240054  FE          260000  FE  54      260054
FE 56       260056  FE  57      260057  FE  58      260058
NI          280000  NI  58      280058  NI  60      280060
NI 61       280061  NI  62      280062  NI  64      280064
MO          420000  MO  92      420092  MO  94      420094
MO 95       420095  MO  96      420096  MO  97      420097
MO 98       420098  MO100       420100  U  235      920235
U 238       920238  PU239       940239  PU240       940240
PU241       940241  PU242       940242  CD          480000
CL          170000  CL  35      170035  CL  37      170037
PB    EN3   820000  U 238WC1    922383  LI   6EN3    30006
LI   7EN3    30007  PU238       940238  U  233EN4   924233
U 234EN4    924234  U 236EN4    924236  PA233EN4    914233
NP237EN4    934237  AM243EN4    954243  CM244EN4    964244
TA181EN4    734181  NB 93EN4    414093  NB 93RCN    410093
MN 55EN4    254055  P   31ENL   154031  GA    ENL   314000
ZR    ENL   404000  EU    ENL   634000  W     ENL   744000
```

```
        THE TYPE CONVERSION TABLE CONTAINS   69 DATA TYPE NAMES



ALPHAMERIC  NUMERIC ALPHAMERIC   NUMERIC ALPHAMERIC   NUMERIC

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   SEDFP        54613   SEDFD        54623
SED2N        50163   SED3N        50173   SEDIC        50053




             ON THE KEDAK-FILE   66 ISOTOPES ARE AVAILABLE :



  NI        NI  58    NI  60    NI  61    NI  62    NI  64    O   16
  U 235     AL  27    C   12    CD        CR        CR  50    CR  52
  CR  53    CR  54    FE        FE  54    FE  56    FE  57    FE  58
  H    2    H   H1    H   01    HE   3    HE   4    MO        MO  92
  MO  94    MO  95    MO  96    MO  97    MO  98    MO100     N
  NA  23    PU239     U  238    H    1    CL        CL  35    CL  37
  PU240     PU242     PU241     PB   EN3  U 238WC1  LI   6EN3 LI   7EN3
  PU238     U 233EN4  U 234EN4  U 236EN4  PA233EN4  NP237EN4  AM243EN4
  CM244EN4  TA181EN4  NB 93EN4  NB 93RCN  MN 55EN4  P   31ENL GA    ENL
  ZR    ENL EU    ENL W     ENL
```

```
FOR NI          19 DATA TYPES ARE AVAILABLE :
 AASTATUS  ISOT1     ISOT2     ISOT3     MUEL      RANGRES   RES
 SGA       SGALP     SGG       SGI       SGIZ      SGN       SGNC
 SGP       SGT       SGTR      SGX       SG2N
THE DATA TYPE SGIZ      HAS 1 FURTHER NAMES AND   12 NAME COMBINATIONS :
 0.133200E+07 0.145200E+07 0.215800E+07 0.228700E+07 0.245800E+07
 0.250200E+07 0.263000E+07 0.277200E+07 0.303500E+07 0.313000E+07
 0.326000E+07 0.352000E+07
THE DATA TYPE SGNC      HAS 1 FURTHER NAMES AND   46 NAME COMBINATIONS :
 0.100000E+05 0.600000E+05 0.130000E+06 0.220000E+06 0.310000E+06
 0.390000E+06 0.470000E+06 0.550000E+06 0.630000E+06 0.720000E+06
 0.800000E+06 0.880000E+06 0.960000E+06 0.104000E+07 0.112000E+07
 0.120000E+07 0.129000E+07 0.136000E+07 0.144000E+07 0.150000E+07
 0.160000E+07 0.170000E+07 0.180000E+07 0.200000E+07 0.220000E+07
 0.225000E+07 0.235000E+07 0.245000E+07 0.250000E+07 0.280000E+07
 0.290000E+07 0.370000E+07 0.399000E+07 0.410000E+07 0.421000E+07
 0.470000E+07 0.500000E+07 0.604000E+07 0.700000E+07 0.800000E+07
 0.900000E+07 0.100000E+08 0.110000E+08 0.120000E+08 0.130000E+08
 0.140000E+08


FOR NI 58        9 DATA TYPES ARE AVAILABLE :
 ISOT1     ISOT2     RANGRES   RES       SGALP     SGP       SG2N
 ST        STD


FOR NI 60        9 DATA TYPES ARE AVAILABLE :
 ISOT1     ISOT2     RANGRES   RES       SGALP     SGP       SG2N
 ST        STD


FOR NI 61        6 DATA TYPES ARE AVAILABLE :
 ISOT1     ISOT2     SGALP     SGP       SG2N      STD


FOR NI 62        8 DATA TYPES ARE AVAILABLE :
 ISOT1     ISOT2     RANGRES   RES       SGALP     SGP       SG2N
 STD


FOR NI 64        6 DATA TYPES ARE AVAILABLE :
 ISOT1     ISOT2     SGALP     SGP       SG2N      STD
```

```
FOR O  16        19 DATA TYPES ARE AVAILABLE :
 AASTATUS   ISOT1      ISOT2      MUEL       RANGRES    RES        SGA
 SGALP      SGD        SGG        SGI        SGIZ       SGN        SGNC
 SGP        SGT        SGTR       SGX        SG2N
THE DATA TYPE SGIZ      HAS 1 FURTHER NAMES AND   24 NAME COMBINATIONS :
 0.605200E+07 0.613100E+07 0.691700E+07 0.711900E+07 0.887200E+07
 0.959700E+07 0.984700E+07 0.103540E+08 0.109520E+08 0.110800E+08
 0.110960E+08 0.112600E+08 0.114400E+08 0.115210E+08 0.116300E+08
 0.120530E+08 0.124420E+08 0.125280E+08 0.127950E+08 0.129670E+08
 0.131500E+08 0.134500E+08 0.137500E+08 0.140500E+08
THE DATA TYPE SGNC      HAS 1 FURTHER NAMES AND  131 NAME COMBINATIONS :
 0.100000E+06 0.126000E+06 0.166000E+06 0.205000E+06 0.297000E+06
 0.326000E+06 0.344000E+06 0.367000E+06 0.385000E+06 0.395000E+06
 0.406000E+06 0.418000E+06 0.429000E+06 0.440000E+06 0.451000E+06
 0.462000E+06 0.473000E+06 0.490000E+06 0.512000E+06 0.539000E+06
 0.572000E+06 0.615000E+06 0.642000E+06 0.680000E+06 0.722000E+06
 0.760000E+06 0.796000E+06 0.817000E+06 0.849000E+06 0.854000E+06
 0.880000E+06 0.912000E+06 0.933000E+06 0.953000E+06 0.974000E+06
 0.995000E+06 0.101600E+07 0.103700E+07 0.107800E+07 0.117600E+07
 0.120500E+07 0.122500E+07 0.124900E+07 0.126100E+07 0.127100E+07
 0.128200E+07 0.128800E+07 0.129400E+07 0.130000E+07 0.130600E+07
 0.131200E+07 0.131800E+07 0.132500E+07 0.133100E+07 0.133800E+07
 0.134600E+07 0.135400E+07 0.136200E+07 0.137000E+07 0.138500E+07
 0.140500E+07 0.143500E+07 0.147500E+07 0.152500E+07 0.158100E+07
 0.160700E+07 0.164300E+07 0.164600E+07 0.164700E+07 0.165100E+07
 0.165200E+07 0.165300E+07 0.165500E+07 0.165700E+07 0.165800E+07
 0.165900E+07 0.166000E+07 0.166100E+07 0.166200E+07 0.166300E+07
 0.166400E+07 0.166500E+07 0.166600E+07 0.166800E+07 0.167000E+07
 0.167200E+07 0.167700E+07 0.168700E+07 0.175000E+07 0.200000E+07
 0.203000E+07 0.215000E+07 0.220000E+07 0.225000E+07 0.234000E+07
 0.256000E+07 0.276000E+07 0.295000E+07 0.300000E+07 0.307000E+07
 0.317000E+07 0.329000E+07 0.335000E+07 0.340000E+07 0.350000E+07
 0.355000E+07 0.361000E+07 0.365000E+07 0.370000E+07 0.375000E+07
 0.380000E+07 0.391000E+07 0.401000E+07 0.405000E+07 0.411000E+07
 0.430000E+07 0.450000E+07 0.485000E+07 0.499000E+07 0.500000E+07
 0.515000E+07 0.566000E+07 0.601000E+07 0.653000E+07 0.700000E+07
 0.711000E+07 0.800000E+07 0.116000E+08 0.141000E+08 0.149200E+08
 0.158300E+08


FOR U 235        29 DATA TYPES ARE AVAILABLE :
 AASTATUS   ALPHA      CHICR      CHIF       ETA        ISOT1      ISOT2
 MUEL       NUE        PLNUE      RANGRES    RES        SGA        SGALP
 SGF        SGG        SGI        SGIZ       SGN        SGNC       SGP
 SGT        SGTR       SGX        SG2N       SG3N       ST         STD
 STGF
THE DATA TYPE SGIZ      HAS 1 FURTHER NAMES AND   10 NAME COMBINATIONS :
 0.100000E+05 0.600000E+05 0.900000E+05 0.200000E+06 0.300000E+06
 0.500000E+06 0.100000E+07 0.150000E+07 0.175000E+07 0.200000E+07
THE DATA TYPE SGNC      HAS 1 FURTHER NAMES AND   43 NAME COMBINATIONS :
 0.100000E+05 0.500000E+05 0.750000E+05 0.100000E+06 0.157000E+06
 0.200000E+06 0.250000E+06 0.300000E+06 0.350000E+06 0.400000E+06
 0.500000E+06 0.550000E+06 0.600000E+06 0.650000E+06 0.720000E+06
 0.770000E+06 0.800000E+06 0.950000E+06 0.980000E+06 0.110000E+07
 0.117000E+07 0.120000E+07 0.125000E+07 0.140000E+07 0.160000E+07
 0.180000E+07 0.200000E+07 0.220000E+07 0.250000E+07 0.300000E+07
 0.400000E+07 0.410000E+07 0.500000E+07 0.600000E+07 0.700000E+07
 0.800000E+07 0.900000E+07 0.100000E+08 0.110000E+08 0.120000E+08
 0.130000E+08 0.140000E+08 0.152000E+08
```

```
FOR AL 27         20 DATA TYPES ARE AVAILABLE :
 AASTATUS  ISOT1     ISOT2     MUEL      RANGRES   RES       SGA
 SGALP     SGG       SGI       SGIZ      SGN       SGNC      SGP
 SGT       SGTR      SGX       SG2N      ST        STD
THE DATA TYPE SGIZ      HAS 1 FURTHER NAMES AND    9 NAME COMBINATIONS :
 0.842000E+06 0.101300E+07 0.221000E+07 0.273000E+07 0.298000E+07
 0.300000E+07 0.368000E+07 0.395000E+07 0.405000E+07
THE DATA TYPE SGNC      HAS 1 FURTHER NAMES AND   36 NAME COMBINATIONS :
 0.100000E+05 0.300000E+05 0.600000E+05 0.130000E+06 0.230000E+06
 0.310000E+06 0.330000E+06 0.400000E+06 0.480000E+06 0.560000E+06
 0.640000E+06 0.730000E+06 0.810000E+06 0.890000E+06 0.970000E+06
 0.105000E+07 0.113000E+07 0.121000E+07 0.129000E+07 0.137000E+07
 0.145000E+07 0.201000E+07 0.250000E+07 0.270000E+07 0.301000E+07
 0.350000E+07 0.370000E+07 0.385000E+07 0.397000E+07 0.410000E+07
 0.420000E+07 0.450000E+07 0.480000E+07 0.500000E+07 0.700000E+07
 0.143000E+08


FOR C  12         19 DATA TYPES ARE AVAILABLE :
 AASTATUS  ISOT1     ISOT2     MUEL      RANGRES   RES       SGA
 SGALP     SGG       SGI       SGIZ      SGI3A     SGN       SGNC
 SGP       SGT       SGTR      SGX       SG2N
THE DATA TYPE SGIZ      HAS 1 FURTHER NAMES AND    5 NAME COMBINATIONS :
 0.443000E+07 0.765000E+07 0.966000E+07 0.108400E+08 0.118200E+08
THE DATA TYPE SGNC      HAS 1 FURTHER NAMES AND   42 NAME COMBINATIONS :
 0.500000E+05 0.100000E+06 0.200000E+06 0.400000E+06 0.600000E+06
 0.800000E+06 0.100000E+07 0.120000E+07 0.140000E+07 0.160000E+07
 0.180000E+07 0.200000E+07 0.205000E+07 0.206000E+07 0.207000E+07
 0.207500E+07 0.208000E+07 0.208500E+07 0.209000E+07 0.210000E+07
 0.212500E+07 0.215000E+07 0.220000E+07 0.228000E+07 0.251000E+07
 0.270000E+07 0.276000E+07 0.295000E+07 0.305000E+07 0.325000E+07
 0.351000E+07 0.376000E+07 0.410000E+07 0.421000E+07 0.470000E+07
 0.500000E+07 0.560000E+07 0.600000E+07 0.630000E+07 0.700000E+07
 0.758000E+07 0.142000E+08


FOR CD            17 DATA TYPES ARE AVAILABLE :
 AASTATUS  ISOT1     ISOT3     MUEL      RANGRES   RES       SGA
 SGALP     SGG       SGI       SGIZ      SGN       SGP       SGT
 SGTR      SGX       SG2N
THE DATA TYPE SGIZ      HAS 1 FURTHER NAMES AND    4 NAME COMBINATIONS :
 0.300000E+06 0.600000E+06 0.120000E+07 0.130000E+07
```

```
FOR CR           19 DATA TYPES ARE AVAILABLE :
 AASTATUS  ISOT1     ISOT2     ISOT3     MUEL      RANGRES   RES
 SGA       SGALP     SGG       SGI       SGIZ      SGN       SGNC
 SGP       SGT       SGTR      SGX       SG2N
THE DATA TYPE SGIZ      HAS 1 FURTHER NAMES AND    8 NAME COMBINATIONS :
 0.565000E+06 0.782000E+06 0.100700E+07 0.143400E+07 0.183500E+07
 0.232700E+07 0.262000E+07 0.296500E+07
THE DATA TYPE SGNC      HAS 1 FURTHER NAMES AND   45 NAME COMBINATIONS :
 0.100000E+05 0.600000E+05 0.160000E+06 0.250000E+06 0.330000E+06
 0.410000E+06 0.500000E+06 0.580000E+06 0.660000E+06 0.730000E+06
 0.820000E+06 0.900000E+06 0.980000E+06 0.106000E+07 0.114000E+07
 0.122000E+07 0.130000E+07 0.138000E+07 0.146000E+07 0.155000E+07
 0.163000E+07 0.170000E+07 0.178000E+07 0.200000E+07 0.235000E+07
 0.245000E+07 0.250000E+07 0.280000E+07 0.290000E+07 0.301000E+07
 0.370000E+07 0.399000E+07 0.410000E+07 0.421000E+07 0.470000E+07
 0.500000E+07 0.604000E+07 0.700000E+07 0.800000E+07 0.900000E+07
 0.100000E+08 0.110000E+08 0.120000E+08 0.130000E+08 0.145000E+08


FOR CR 50          9 DATA TYPES ARE AVAILABLE :
 ISOT1     ISOT2     RANGRES   RES       SGALP     SGP       SG2N
 ST        STD


FOR CR 52          9 DATA TYPES ARE AVAILABLE :
 ISOT1     ISOT2     RANGRES   RES       SGALP     SGP       SG2N
 ST        STD


FOR CR 53          9 DATA TYPES ARE AVAILABLE :
 ISOT1     ISOT2     RANGRES   RES       SGALP     SGP       SG2N
 ST        STD


FOR CR 54          9 DATA TYPES ARE AVAILABLE :
 ISOT1     ISOT2     RANGRES   RES       SGALP     SGP       SG2N
 ST        STD
```

```
FOR FE          19 DATA TYPES ARE AVAILABLE :
 AASTATUS  ISOT1     ISOT2     ISOT3     MUEL      RANGRES   RES
 SGA       SGALP     SGG       SGI       SGIZ      SGN       SGNC
 SGP       SGT       SGTR      SGX       SG2N
THE DATA TYPE SGIZ      HAS 1 FURTHER NAMES AND   10 NAME COMBINATIONS :
 0.845000E+06 0.140800E+07 0.208000E+07 0.265500E+07 0.293600E+07
 0.311800E+07 0.336700E+07 0.359900E+07 0.382500E+07 0.403800E+07
THE DATA TYPE SGNC      HAS 1 FURTHER NAMES AND   45 NAME COMBINATIONS :
 0.100000E+05 0.600000E+05 0.100000E+06 0.200000E+06 0.300000E+06
 0.400000E+06 0.500000E+06 0.600000E+06 0.700000E+06 0.800000E+06
 0.900000E+06 0.100000E+07 0.110000E+07 0.120000E+07 0.130000E+07
 0.140000E+07 0.150000E+07 0.160000E+07 0.170000E+07 0.180000E+07
 0.195000E+07 0.201000E+07 0.220000E+07 0.225000E+07 0.235000E+07
 0.245000E+07 0.250000E+07 0.280000E+07 0.290000E+07 0.301000E+07
 0.370000E+07 0.399000E+07 0.410000E+07 0.421000E+07 0.470000E+07
 0.500000E+07 0.604000E+07 0.700000E+07 0.800000E+07 0.900000E+07
 0.100000E+08 0.110000E+08 0.120000E+08 0.130000E+08 0.145000E+08


FOR FE 54        9 DATA TYPES ARE AVAILABLE :
 ISOT1     ISOT2     RANGRES   RES       SGALP     SGP       SG2N
 ST        STD


FOR FE 56        9 DATA TYPES ARE AVAILABLE :
 ISOT1     ISOT2     RANGRES   RES       SGALP     SGP       SG2N
 ST        STD


FOR FE 57        9 DATA TYPES ARE AVAILABLE :
 ISOT1     ISOT2     RANGRES   RES       SGALP     SGP       SG2N
 ST        STD


FOR FE 58        5 DATA TYPES ARE AVAILABLE :
 ISOT1     ISOT2     SGALP     SGP       STD
```

```
FOR H   2         15 DATA TYPES ARE AVAILABLE :
 AASTATUS  ISOT1     ISOT2     MUEL      SGA       SGALP     SGG
 SGI       SGN       SGNC      SGP       SGT       SGTR      SGX
 SG2N
THE DATA TYPE SGNC      HAS 1 FURTHER NAMES AND   14 NAME COMBINATIONS :
 0.500000E+05 0.100000E+06 0.200000E+06 0.220000E+06 0.500000E+06
 0.750000E+06 0.100000E+07 0.150000E+07 0.200000E+07 0.250000E+07
 0.327000E+07 0.450000E+07 0.550000E+07 0.141000E+08


FOR H  H1         13 DATA TYPES ARE AVAILABLE :
 AASTATUS  MUEL      SGA       SGALP     SGG       SGI       SGN
 SGNC      SGP       SGT       SGTR      SGX       SG2N
THE DATA TYPE SGNC      HAS 1 FURTHER NAMES AND   19 NAME COMBINATIONS :
 0.500000E+05 0.100000E+06 0.200000E+06 0.400000E+06 0.600000E+06
 0.800000E+06 0.100000E+07 0.200000E+07 0.300000E+07 0.400000E+07
 0.500000E+07 0.600000E+07 0.700000E+07 0.800000E+07 0.900000E+07
 0.100000E+08 0.120000E+08 0.140000E+08 0.160000E+08


FOR H  01         13 DATA TYPES ARE AVAILABLE :
 AASTATUS  MUEL      SGA       SGALP     SGG       SGI       SGN
 SGNC      SGP       SGT       SGTR      SGX       SG2N
THE DATA TYPE SGNC      HAS 1 FURTHER NAMES AND   19 NAME COMBINATIONS :
 0.500000E+05 0.100000E+06 0.200000E+06 0.400000E+06 0.600000E+06
 0.800000E+06 0.100000E+07 0.200000E+07 0.300000E+07 0.400000E+07
 0.500000E+07 0.600000E+07 0.700000E+07 0.800000E+07 0.900000E+07
 0.100000E+08 0.120000E+08 0.140000E+08 0.160000E+08


FOR HE  3         14 DATA TYPES ARE AVAILABLE :
 AASTATUS  ISOT1     ISOT2     MUEL      SGA       SGD       SGG
 SGI       SGN       SGNC      SGP       SGT       SGTR      SGX
THE DATA TYPE SGNC      HAS 1 FURTHER NAMES AND    8 NAME COMBINATIONS :
 0.100000E+07 0.200000E+07 0.260000E+07 0.350000E+07 0.500000E+07
 0.600000E+07 0.800000E+07 0.150000E+08


FOR HE  4         12 DATA TYPES ARE AVAILABLE :
 AASTATUS  ISOT1     ISOT2     MUEL      SGA       SGG       SGI
 SGN       SGNC      SGT       SGTR      SGX
THE DATA TYPE SGNC      HAS 1 FURTHER NAMES AND   26 NAME COMBINATIONS :
 0.100000E+06 0.200000E+06 0.300000E+06 0.400000E+06 0.600000E+06
 0.750000E+06 0.865000E+06 0.100000E+07 0.120000E+07 0.140000E+07
 0.170000E+07 0.200000E+07 0.240000E+07 0.261000E+07 0.273000E+07
 0.302000E+07 0.405000E+07 0.453000E+07 0.554000E+07 0.597000E+07
 0.650000E+07 0.796000E+07 0.100000E+08 0.120000E+08 0.143000E+08
 0.147000E+08
```

```
FOR MO          19 DATA TYPES ARE AVAILABLE :
 AASTATUS   ISOT1      ISOT2      ISOT3      MUEL       RANGRES    RES
 SGA        SGALP      SGG        SGI        SGIZ       SGN        SGNC
 SGP        SGT        SGTR       SGX        SG2N
THE DATA TYPE SGIZ      HAS 1 FURTHER NAMES AND    8 NAME COMBINATIONS :
 0.203000E+06 0.530000E+06 0.780000E+06 0.930000E+06 0.110000E+07
 0.126000E+07 0.150000E+07 0.186000E+07
THE DATA TYPE SGNC      HAS 1 FURTHER NAMES AND   39 NAME COMBINATIONS :
 0.100000E+05 0.300000E+05 0.600000E+05 0.140000E+06 0.230000E+06
 0.330000E+06 0.400000E+06 0.450000E+06 0.480000E+06 0.510000E+06
 0.550000E+06 0.600000E+06 0.640000E+06 0.700000E+06 0.730000E+06
 0.800000E+06 0.850000E+06 0.890000E+06 0.970000E+06 0.100000E+07
 0.105000E+07 0.113000E+07 0.121000E+07 0.129000E+07 0.137000E+07
 0.200000E+07 0.290000E+07 0.370000E+07 0.410000E+07 0.500000E+07
 0.604000E+07 0.700000E+07 0.800000E+07 0.900000E+07 0.100000E+08
 0.110000E+08 0.120000E+08 0.130000E+08 0.140000E+08


FOR MO 92        9 DATA TYPES ARE AVAILABLE :
 ISOT1      ISOT2      RANGRES    RES        SGALP      SGP        SG2N
 ST         STD


FOR MO 94        9 DATA TYPES ARE AVAILABLE :
 ISOT1      ISOT2      RANGRES    RES        SGALP      SGP        SG2N
 ST         STD


FOR MO 95        9 DATA TYPES ARE AVAILABLE :
 ISOT1      ISOT2      RANGRES    RES        SGALP      SGP        SG2N
 ST         STD


FOR MO 96        9 DATA TYPES ARE AVAILABLE :
 ISOT1      ISOT2      RANGRES    RES        SGALP      SGP        SG2N
 ST         STD


FOR MO 97        9 DATA TYPES ARE AVAILABLE :
 ISOT1      ISOT2      RANGRES    RES        SGALP      SGP        SG2N
 ST         STD


FOR MO 98        9 DATA TYPES ARE AVAILABLE :
 ISOT1      ISOT2      RANGRES    RES        SGALP      SGP        SG2N
 ST         STD


FOR MO100        9 DATA TYPES ARE AVAILABLE :
 ISOT1      ISOT2      RANGRES    RES        SGALP      SGP        SG2N
 ST         STD
```

```
FOR N            2 DATA TYPES ARE AVAILABLE :
 AASTATUS  SGNC
THE DATA TYPE SGNC      HAS 1 FURTHER NAMES AND   41 NAME COMBINATIONS :
 0.100000E+06 0.300000E+06 0.500000E+06 0.800000E+06 0.970000E+06
 0.108200E+07 0.111000E+07 0.112000E+07 0.113000E+07 0.116000E+07
 0.128000E+07 0.135000E+07 0.137700E+07 0.140100E+07 0.154000E+07
 0.159500E+07 0.168200E+07 0.175600E+07 0.177900E+07 0.179600E+07
 0.207000E+07 0.225000E+07 0.236000E+07 0.307000E+07 0.351000E+07
 0.405000E+07 0.430000E+07 0.450000E+07 0.485000E+07 0.499000E+07
 0.515000E+07 0.556000E+07 0.602000E+07 0.653000E+07 0.700000E+07
 0.711000E+07 0.800000E+07 0.116000E+08 0.140000E+08 0.149200E+08
 0.158300E+08


FOR NA 23       20 DATA TYPES ARE AVAILABLE :
 AASTATUS  ISOT1     ISOT2     MUEL      RANGRES   RES       SGA
 SGALP     SGG       SGI       SGIZ      SGN       SGNC      SGP
 SGT       SGTR      SGX       SG2N      ST        STD
THE DATA TYPE SGIZ      HAS 1 FURTHER NAMES AND    7 NAME COMBINATIONS :
 0.439000E+06 0.207800E+07 0.239300E+07 0.264100E+07 0.270500E+07
 0.298300E+07 0.368000E+07
THE DATA TYPE SGNC      HAS 1 FURTHER NAMES AND   63 NAME COMBINATIONS :
 0.100000E+05 0.300000E+05 0.600000E+05 0.140000E+06 0.210000E+06
 0.220000E+06 0.235000E+06 0.245000E+06 0.285000E+06 0.305000E+06
 0.325000E+06 0.355000E+06 0.375000E+06 0.390000E+06 0.400000E+06
 0.410000E+06 0.435000E+06 0.445000E+06 0.475000E+06 0.520000E+06
 0.545000E+06 0.575000E+06 0.605000E+06 0.615000E+06 0.625000E+06
 0.635000E+06 0.670000E+06 0.700000E+06 0.710000E+06 0.720000E+06
 0.755000E+06 0.765000E+06 0.775000E+06 0.795000E+06 0.820000E+06
 0.900000E+06 0.100000E+07 0.110000E+07 0.120000E+07 0.130000E+07
 0.140000E+07 0.150000E+07 0.160000E+07 0.170000E+07 0.180000E+07
 0.190000E+07 0.200000E+07 0.210000E+07 0.220000E+07 0.251500E+07
 0.270000E+07 0.301000E+07 0.350000E+07 0.370000E+07 0.385000E+07
 0.397000E+07 0.410000E+07 0.420000E+07 0.450000E+07 0.480000E+07
 0.500000E+07 0.700000E+07 0.143000E+08


FOR PU239       29 DATA TYPES ARE AVAILABLE :
 AASTATUS  ALPHA     CHICR     CHIF      ETA       ISOT1     ISOT2
 MUEL      NUE       PLNUE     RANGRES   RES       SGA       SGALP
 SGF       SGG       SGI       SGIZ      SGN       SGNC      SGP
 SGT       SGTR      SGX       SG2N      SG3N      ST        STD
 STGF
THE DATA TYPE SGIZ      HAS 1 FURTHER NAMES AND    7 NAME COMBINATIONS :
 0.800000E+04 0.570000E+05 0.760000E+05 0.164000E+06 0.286000E+06
 0.331000E+06 0.392000E+06
THE DATA TYPE SGNC      HAS 1 FURTHER NAMES AND   43 NAME COMBINATIONS :
 0.100000E+05 0.500000E+05 0.750000E+05 0.100000E+06 0.157000E+06
 0.200000E+06 0.250000E+06 0.300000E+06 0.350000E+06 0.400000E+06
 0.500000E+06 0.550000E+06 0.600000E+06 0.650000E+06 0.720000E+06
 0.770000E+06 0.800000E+06 0.950000E+06 0.980000E+06 0.110000E+07
 0.117000E+07 0.120000E+07 0.125000E+07 0.140000E+07 0.160000E+07
 0.180000E+07 0.200000E+07 0.220000E+07 0.250000E+07 0.300000E+07
 0.400000E+07 0.410000E+07 0.500000E+07 0.600000E+07 0.700000E+07
 0.800000E+07 0.900000E+07 0.100000E+08 0.110000E+08 0.120000E+08
 0.130000E+08 0.140000E+08 0.152000E+08
```

```
FOR U 238       27 DATA TYPES ARE AVAILABLE :
 AASTATUS   CHICR      CHIF       ISOT1      ISOT2      MUEL       NUE
 PLNUE      RANGRES    RES        SGA        SGALP      SGF        SGG
 SGI        SGIZ       SGIZC      SGN        SGNC       SGP        SGT
 SGTR       SGX        SG2N       SG3N       ST         STD
THE DATA TYPE SGIZ     HAS 1 FURTHER NAMES AND   26 NAME COMBINATIONS :
 0.450000E+05 0.146000E+06 0.308000E+06 0.680000E+06 0.732000E+06
 0.827000E+06 0.930000E+06 0.967000E+06 0.100000E+07 0.104100E+07
 0.106000E+07 0.112000E+07 0.116000E+07 0.122000E+07 0.127000E+07
 0.130000E+07 0.136100E+07 0.140900E+07 0.143700E+07 0.147000E+07
 0.162500E+07 0.187500E+07 0.195000E+07 0.295000E+07 0.395000E+07
 0.495000E+07
THE DATA TYPE SGNC     HAS 1 FURTHER NAMES AND   42 NAME COMBINATIONS :
 0.100000E+05 0.500000E+05 0.750000E+05 0.100000E+06 0.157000E+06
 0.200000E+06 0.250000E+06 0.300000E+06 0.350000E+06 0.400000E+06
 0.500000E+06 0.550000E+06 0.600000E+06 0.650000E+06 0.720000E+06
 0.770000E+06 0.800000E+06 0.950000E+06 0.980000E+06 0.110000E+07
 0.117000E+07 0.120000E+07 0.125000E+07 0.140000E+07 0.160000E+07
 0.180000E+07 0.200000E+07 0.220000E+07 0.250000E+07 0.300000E+07
 0.400000E+07 0.410000E+07 0.500000E+07 0.600000E+07 0.700000E+07
 0.800000E+07 0.900000E+07 0.100000E+08 0.110000E+08 0.120000E+08
 0.130000E+08 0.140000E+08


FOR H   1        2 DATA TYPES ARE AVAILABLE :
 ISOT1      ISOT2


FOR CL          17 DATA TYPES ARE AVAILABLE :
 AASTATUS   ISOT1      ISOT2      ISOT3      MUEL       RANGRES    RES
 SGA        SGALP      SGG        SGI        SGN        SGP        SGT
 SGTR       SGX        SG2N


FOR CL  35       2 DATA TYPES ARE AVAILABLE :
 ISOT1      ISOT2


FOR CL  37       2 DATA TYPES ARE AVAILABLE :
 ISOT1      ISOT2
```

```
FOR PU240        25 DATA TYPES ARE AVAILABLE :
 AASTATUS   ALPHA      CHICR      ETA        ISOT1      ISOT2      MUEL
 NUE        PLNUE      RANGRES    RES        SGA        SGF        SGG
 SGI        SGIZ       SGN        SGNC       SGT        SGTR       SGX
 SG2N       SG3N       ST         STD        STGF
THE DATA TYPE SGIZ       HAS 1 FURTHER NAMES AND   20 NAME COMBINATIONS :
 0.430000E+05 0.142000E+06 0.294000E+06 0.597000E+06 0.649000E+06
 0.742000E+06 0.861000E+06 0.900000E+06 0.938000E+06 0.959000E+06
 0.100200E+07 0.103100E+07 0.103800E+07 0.109100E+07 0.111600E+07
 0.113700E+07 0.116100E+07 0.130800E+07 0.141100E+07 0.143800E+07
THE DATA TYPE SGNC       HAS 1 FURTHER NAMES AND   70 NAME COMBINATIONS :
 0.100000E+04 0.300000E+04 0.500000E+04 0.800000E+04 0.150000E+05
 0.203000E+05 0.303000E+05 0.430000E+05 0.499000E+05 0.552000E+05
 0.610000E+05 0.745000E+05 0.823000E+05 0.910000E+05 0.111000E+06
 0.136000E+06 0.142000E+06 0.166000E+06 0.202000E+06 0.247000E+06
 0.294000E+06 0.400000E+06 0.450000E+06 0.550000E+06 0.597000E+06
 0.649000E+06 0.705000E+06 0.742000E+06 0.805000E+06 0.861000E+06
 0.900000E+06 0.938000E+06 0.959000E+06 0.100200E+07 0.103100E+07
 0.103800E+07 0.109100E+07 0.111600E+07 0.113700E+07 0.116100E+07
 0.120500E+07 0.130800E+07 0.141100E+07 0.143800E+07 0.150000E+07
 0.165000E+07 0.183000E+07 0.202000E+07 0.223000E+07 0.247000E+07
 0.273000E+07 0.301000E+07 0.333000E+07 0.368000E+07 0.407000E+07
 0.449000E+07 0.497000E+07 0.549000E+07 0.607000E+07 0.648000E+07
 0.670000E+07 0.741000E+07 0.819000E+07 0.905000E+07 0.100000E+08
 0.111000E+08 0.121000E+08 0.122000E+08 0.135000E+08 0.150000E+08


FOR PU242        24 DATA TYPES ARE AVAILABLE :
 AASTATUS   CHICR      ISOT1      ISOT2      MUEL       NUE        PLNUE
 RANGRES    RES        SGA        SGF        SGG        SGI        SGIZ
 SGN        SGNC       SGT        SGTR       SGX        SG2N       SG3N
 ST         STD        STGF
THE DATA TYPE SGIZ       HAS 1 FURTHER NAMES AND   17 NAME COMBINATIONS :
 0.440000E+05 0.146000E+06 0.294000E+06 0.597000E+06 0.649000E+06
 0.742000E+06 0.956000E+06 0.995000E+06 0.100200E+07 0.103100E+07
 0.103800E+07 0.109100E+07 0.110700E+07 0.116100E+07 0.130800E+07
 0.141100E+07 0.143800E+07
THE DATA TYPE SGNC       HAS 1 FURTHER NAMES AND   74 NAME COMBINATIONS :
 0.200000E+03 0.300000E+03 0.400000E+03 0.600000E+03 0.800000E+03
 0.100000E+04 0.300000E+04 0.500000E+04 0.800000E+04 0.150000E+05
 0.203000E+05 0.303000E+05 0.440000E+05 0.499000E+05 0.552000E+05
 0.610000E+05 0.745000E+05 0.823000E+05 0.910000E+05 0.111000E+06
 0.136000E+06 0.146000E+06 0.166000E+06 0.202000E+06 0.247000E+06
 0.294000E+06 0.400000E+06 0.450000E+06 0.550000E+06 0.597000E+06
 0.649000E+06 0.705000E+06 0.742000E+06 0.805000E+06 0.861000E+06
 0.900000E+06 0.938000E+06 0.956000E+06 0.995000E+06 0.100200E+07
 0.103100E+07 0.103800E+07 0.109100E+07 0.110700E+07 0.116100E+07
 0.130800E+07 0.141100E+07 0.143800E+07 0.150000E+07 0.165000E+07
 0.183000E+07 0.202000E+07 0.223000E+07 0.247000E+07 0.273000E+07
 0.301000E+07 0.333000E+07 0.368000E+07 0.407000E+07 0.449000E+07
 0.497000E+07 0.549000E+07 0.607000E+07 0.629000E+07 0.670000E+07
 0.741000E+07 0.819000E+07 0.905000E+07 0.100000E+08 0.111000E+08
 0.117000E+08 0.122000E+08 0.135000E+08 0.150000E+08
```

```
FOR PU241        25 DATA TYPES ARE AVAILABLE :
 AASTATUS  ALPHA     CHICR     ETA       ISOT1     ISOT2     MUEL
 NUE       PLNUE     RANGRES   RES       SGA       SGF       SGG
 SGI       SGIZ      SGN       SGNC      SGT       SGTR      SGX
 SG2N      SG3N      ST        STD       STGF
THE DATA TYPE SGIZ      HAS 1 FURTHER NAMES AND    19 NAME COMBINATIONS :
 0.400000E+05 0.920000E+05 0.163000E+06 0.167000E+06 0.169000E+06
 0.172000E+06 0.230000E+06 0.235000E+06 0.235100E+06 0.244000E+06
 0.296000E+06 0.334000E+06 0.444000E+06 0.499000E+06 0.568000E+06
 0.809000E+06 0.835000E+06 0.875000E+06 0.931000E+06
THE DATA TYPE SGNC      HAS 1 FURTHER NAMES AND    72 NAME COMBINATIONS :
 0.100000E+03 0.150000E+03 0.200000E+03 0.300000E+03 0.400000E+03
 0.600000E+03 0.800000E+03 0.100000E+04 0.300000E+04 0.500000E+04
 0.800000E+04 0.150000E+05 0.203000E+05 0.303000E+05 0.400000E+05
 0.499000E+05 0.552000E+05 0.610000E+05 0.745000E+05 0.823000E+05
 0.920000E+05 0.100000E+06 0.136000E+06 0.163000E+06 0.167000E+06
 0.169000E+06 0.172000E+06 0.200000E+06 0.230000E+06 0.235000E+06
 0.244000E+06 0.296000E+06 0.334000E+06 0.400000E+06 0.444000E+06
 0.499000E+06 0.568000E+06 0.650000E+06 0.809000E+06 0.835000E+06
 0.875000E+06 0.931000E+06 0.994000E+06 0.100000E+07 0.130000E+07
 0.150000E+07 0.183000E+07 0.202000E+07 0.223000E+07 0.247000E+07
 0.273000E+07 0.301000E+07 0.333000E+07 0.368000E+07 0.407000E+07
 0.449000E+07 0.497000E+07 0.541000E+07 0.607000E+07 0.629000E+07
 0.670000E+07 0.741000E+07 0.819000E+07 0.905000E+07 0.100000E+08
 0.110000E+08 0.119000E+08 0.122000E+08 0.128000E+08 0.135000E+08
 0.142000E+08 0.150000E+08



FOR PB    EN3     16 DATA TYPES ARE AVAILABLE :
 ISOT1     ISOT2     LEGNC     MUEL      SGA       SGG       SGI
 SGIZ      SGIZC     SGN       SGNC      SGT       SGTR      SGX
 SG2N      SG3N
THE DATA TYPE LEGNC     HAS 2 FURTHER NAMES AND    23 NAME COMBINATIONS :
 0.100000E-04 0.100000E+01 0.100000E+05 0.100000E+01 0.250000E+06
 0.200000E+01 0.225000E+07 0.400000E+01 0.300000E+07 0.500000E+01
 0.350000E+07 0.600000E+01 0.500000E+07 0.900000E+01 0.550000E+07
 0.900000E+01 0.600000E+07 0.100000E+02 0.700000E+07 0.100000E+02
 0.800000E+07 0.110000E+02 0.900000E+07 0.110000E+02 0.100000E+08
 0.120000E+02 0.110000E+08 0.120000E+02 0.120000E+08 0.120000E+02
 0.130000E+08 0.130000E+02 0.140000E+08 0.130000E+02 0.150000E+08
 0.140000E+02 0.160000E+08 0.140000E+02 0.170000E+08 0.140000E+02
 0.180000E+08 0.150000E+02 0.190000E+08 0.150000E+02 0.200000E+08
 0.150000E+02
THE DATA TYPE SGIZ      HAS 1 FURTHER NAMES AND    35 NAME COMBINATIONS :
 0.570000E+06 0.803000E+06 0.898000E+06 0.117500E+07 0.134100E+07
 0.146200E+07 0.163300E+07 0.168200E+07 0.176200E+07 0.199800E+07
 0.216000E+07 0.234000E+07 0.238500E+07 0.261500E+07 0.262400E+07
 0.263400E+07 0.278300E+07 0.301700E+07 0.305700E+07 0.319800E+07
 0.325000E+07 0.338200E+07 0.345300E+07 0.347500E+07 0.356000E+07
 0.370800E+07 0.375000E+07 0.385400E+07 0.392000E+07 0.398900E+07
 0.407600E+07 0.412500E+07 0.420000E+07 0.428800E+07 0.433900E+07
THE DATA TYPE SGNC      HAS 1 FURTHER NAMES AND    23 NAME COMBINATIONS :
 0.100000E-04 0.100000E+05 0.250000E+06 0.225000E+07 0.300000E+07
 0.350000E+07 0.500000E+07 0.550000E+07 0.600000E+07 0.700000E+07
 0.800000E+07 0.900000E+07 0.100000E+08 0.110000E+08 0.120000E+08
 0.130000E+08 0.140000E+08 0.150000E+08 0.160000E+08 0.170000E+08
 0.180000E+08 0.190000E+08 0.200000E+08
```

```
FOR U 238WC1      3 DATA TYPES ARE AVAILABLE :
 SGG          SGI          SGIZ
THE DATA TYPE SGIZ        HAS 1 FURTHER NAMES AND    24 NAME COMBINATIONS :
 0.447000E+05 0.148000E+06 0.310000E+06 0.680000E+06 0.732000E+06
 0.838000E+06 0.939000E+06 0.968000E+06 0.100600E+07 0.104700E+07
 0.107600E+07 0.112300E+07 0.115000E+07 0.119000E+07 0.121000E+07
 0.124600E+07 0.127200E+07 0.131300E+07 0.136100E+07 0.140100E+07
 0.143700E+07 0.147000E+07 0.150000E+07 0.175000E+07


FOR LI   6EN3      17 DATA TYPES ARE AVAILABLE :
 ISOT1        ISOT2        LEGNC        MUEL         SEDIC        SGA          SGALP
 SGG          SGI          SGIZ         SGIZC        SGN          SGNC         SGP
 SGT          SGTR         SG2NA
THE DATA TYPE LEGNC       HAS 2 FURTHER NAMES AND    35 NAME COMBINATIONS :
 0.100000E-04 0.100000E+01 0.100000E+05 0.100000E+01 0.120000E+06
 0.300000E+01 0.207000E+06 0.300000E+01 0.225000E+06 0.400000E+01
 0.235000E+06 0.400000E+01 0.245000E+06 0.400000E+01 0.255000E+06
 0.400000E+01 0.265000E+06 0.400000E+01 0.275000E+06 0.400000E+01
 0.307000E+06 0.300000E+01 0.357000E+06 0.300000E+01 0.407000E+06
 0.300000E+01 0.507000E+06 0.200000E+01 0.630000E+06 0.400000E+01
 0.830000E+06 0.400000E+01 0.104000E+07 0.400000E+01 0.124000E+07
 0.400000E+01 0.144000E+07 0.400000E+01 0.175000E+07 0.400000E+01
 0.225000E+07 0.400000E+01 0.250000E+07 0.400000E+01 0.350000E+07
 0.400000E+01 0.400000E+07 0.400000E+01 0.483000E+07 0.400000E+01
 0.574000E+07 0.400000E+01 0.750000E+07 0.400000E+01 0.100000E+08
 0.800000E+01 0.120000E+08 0.800000E+01 0.130000E+08 0.800000E+01
 0.140000E+08 0.800000E+01 0.150000E+08 0.800000E+01 0.160000E+08
 0.800000E+01 0.180000E+08 0.800000E+01 0.200000E+08 0.800000E+01
THE DATA TYPE SEDIC       HAS 1 FURTHER NAMES AND     3 NAME COMBINATIONS :
 0.171800E+07 0.410000E+07 0.200000E+08
THE DATA TYPE SGIZ        HAS 1 FURTHER NAMES AND     1 NAME COMBINATIONS :
 0.356000E+07
THE DATA TYPE SGNC        HAS 1 FURTHER NAMES AND    35 NAME COMBINATIONS :
 0.100000E-04 0.100000E+05 0.120000E+06 0.207000E+06 0.225000E+06
 0.235000E+06 0.245000E+06 0.255000E+06 0.265000E+06 0.275000E+06
 0.307000E+06 0.357000E+06 0.407000E+06 0.507000E+06 0.630000E+06
 0.830000E+06 0.104000E+07 0.124000E+07 0.144000E+07 0.175000E+07
 0.225000E+07 0.250000E+07 0.350000E+07 0.400000E+07 0.483000E+07
 0.574000E+07 0.750000E+07 0.100000E+08 0.120000E+08 0.130000E+08
 0.140000E+08 0.150000E+08 0.160000E+08 0.180000E+08 0.200000E+08
```

```
FOR LI  7EN3     18 DATA TYPES ARE AVAILABLE :
 ISOT1      ISOT2      LEGNC      MUEL       SEDIC      SED2N      SGA
 SGD        SGG        SGI        SGIZ       SGIZC      SGN        SGNC
 SGT        SGTR       SG2N       SG2NA
THE DATA TYPE LEGNC     HAS 2 FURTHER NAMES AND   16 NAME COMBINATIONS :
 0.100000E-04 0.100000E+01 0.100000E+05 0.100000E+01 0.205000E+06
 0.500000E+01 0.229000E+06 0.700000E+01 0.255000E+06 0.900000E+01
 0.259000E+06 0.900000E+01 0.275000E+06 0.900000E+01 0.600000E+06
 0.900000E+01 0.150000E+07 0.700000E+01 0.335000E+07 0.900000E+01
 0.400000E+07 0.700000E+01 0.515000E+07 0.900000E+01 0.636000E+07
 0.900000E+01 0.754000E+07 0.900000E+01 0.140000E+08 0.900000E+01
 0.150000E+08 0.900000E+01
THE DATA TYPE SEDIC     HAS 1 FURTHER NAMES AND    4 NAME COMBINATIONS :
 0.282100E+07 0.580000E+07 0.800000E+07 0.150000E+08
THE DATA TYPE SED2N     HAS 1 FURTHER NAMES AND    3 NAME COMBINATIONS :
 0.830000E+07 0.116500E+08 0.150000E+08
THE DATA TYPE SGIZ      HAS 1 FURTHER NAMES AND    1 NAME COMBINATIONS :
 0.478000E+06
THE DATA TYPE SGNC      HAS 1 FURTHER NAMES AND   16 NAME COMBINATIONS :
 0.100000E-04 0.100000E+05 0.205000E+06 0.229000E+06 0.255000E+06
 0.259000E+06 0.275000E+06 0.600000E+06 0.150000E+07 0.335000E+07
 0.400000E+07 0.515000E+07 0.636000E+07 0.754000E+07 0.140000E+08
 0.150000E+08



FOR PU238        26 DATA TYPES ARE AVAILABLE :
 AASTATUS   ALPHA      CHICR      ETA        ISOT1      ISOT2      MUEL
 NUE        PLNUE      RANGRES    RES        SGA        SGF        SGG
 SGI        SGIZ       SGN        SGNC       SGT        SGTR       SGX
 SG2N       SG3N       ST         STD        STGF
THE DATA TYPE SGIZ      HAS 1 FURTHER NAMES AND   19 NAME COMBINATIONS :
 0.440800E+05 0.145960E+06 0.303600E+06 0.605180E+06 0.661450E+06
 0.941500E+06 0.962770E+06 0.968900E+06 0.983000E+06 0.985460E+06
 0.102850E+07 0.106990E+07 0.108260E+07 0.120270E+07 0.122860E+07
 0.126420E+07 0.144730E+07 0.162140E+07 0.163660E+07
THE DATA TYPE SGNC      HAS 1 FURTHER NAMES AND  126 NAME COMBINATIONS :
 0.465000E+03 0.502000E+03 0.555000E+03 0.613000E+03 0.677000E+03
 0.749000E+03 0.827000E+03 0.914000E+03 0.101000E+04 0.112000E+04
 0.123000E+04 0.136000E+04 0.151000E+04 0.167000E+04 0.184000E+04
 0.203000E+04 0.225000E+04 0.249000E+04 0.275000E+04 0.304000E+04
 0.335000E+04 0.371000E+04 0.410000E+04 0.453000E+04 0.500000E+04
 0.553000E+04 0.611000E+04 0.676000E+04 0.747000E+04 0.825000E+04
 0.912000E+04 0.101000E+05 0.111000E+05 0.123000E+05 0.136000E+05
 0.150000E+05 0.166000E+05 0.184000E+05 0.203000E+05 0.224000E+05
 0.248000E+05 0.274000E+05 0.303000E+05 0.335000E+05 0.370000E+05
 0.409000E+05 0.442000E+05 0.499000E+05 0.552000E+05 0.610000E+05
 0.674000E+05 0.745000E+05 0.823000E+05 0.910000E+05 0.101000E+06
 0.111000E+06 0.123000E+06 0.136000E+06 0.146000E+06 0.166000E+06
 0.183000E+06 0.202000E+06 0.224000E+06 0.247000E+06 0.273000E+06
 0.304000E+06 0.334000E+06 0.369000E+06 0.408000E+06 0.450000E+06
 0.516000E+06 0.550000E+06 0.607000E+06 0.664000E+06 0.743000E+06
 0.821000E+06 0.850000E+06 0.907000E+06 0.945000E+06 0.966000E+06
 0.973000E+06 0.987000E+06 0.989000E+06 0.100000E+07 0.103200E+07
 0.107400E+07 0.108700E+07 0.120700E+07 0.123300E+07 0.126900E+07
 0.135000E+07 0.145300E+07 0.150000E+07 0.162800E+07 0.164300E+07
 0.170000E+07 0.183000E+07 0.202000E+07 0.223000E+07 0.247000E+07
 0.273000E+07 0.301000E+07 0.333000E+07 0.368000E+07 0.407000E+07
 0.449000E+07 0.460000E+07 0.480000E+07 0.497000E+07 0.549000E+07
 0.607000E+07 0.640000E+07 0.670000E+07 0.703000E+07 0.741000E+07
 0.760000E+07 0.819000E+07 0.905000E+07 0.100000E+08 0.111000E+08
 0.122000E+08 0.130000E+08 0.135000E+08 0.140000E+08 0.145000E+08
 0.150000E+08
```

```
FOR U 233EN4    22 DATA TYPES ARE AVAILABLE :
 ISOT1       ISOT2       LEGNC       MUEL        NUE         RANGRES     RES
 SEDIC       SED2N       SED3N       SGA         SGF         SGG         SGI
 SGIZ        SGIZC       SGN         SGNC        SGT         SGTR        SG2N
 SG3N
THE DATA TYPE LEGNC     HAS 2 FURTHER NAMES AND   43 NAME COMBINATIONS :
 0.100000E-04 0.100000E+01 0.100000E+01 0.100000E+01 0.100000E+03
 0.100000E+01 0.500000E+04 0.100000E+01 0.100000E+05 0.600000E+01
 0.200000E+05 0.600000E+01 0.500000E+05 0.800000E+01 0.600000E+05
 0.800000E+01 0.800000E+05 0.800000E+01 0.100000E+06 0.800000E+01
 0.120000E+06 0.100000E+02 0.140000E+06 0.100000E+02 0.160000E+06
 0.100000E+02 0.180000E+06 0.100000E+02 0.200000E+06 0.100000E+02
 0.250000E+06 0.100000E+02 0.300000E+06 0.100000E+02 0.350000E+06
 0.100000E+02 0.400000E+06 0.120000E+02 0.500000E+06 0.120000E+02
 0.610000E+06 0.140000E+02 0.650000E+06 0.120000E+02 0.700000E+06
 0.140000E+02 0.800000E+06 0.140000E+02 0.900000E+06 0.140000E+02
 0.910000E+06 0.140000E+02 0.950000E+06 0.140000E+02 0.100000E+07
 0.140000E+02 0.120000E+07 0.140000E+02 0.140000E+07 0.160000E+02
 0.160000E+07 0.160000E+02 0.180000E+07 0.160000E+02 0.200000E+07
 0.160000E+02 0.250000E+07 0.180000E+02 0.300000E+07 0.200000E+02
 0.350000E+07 0.200000E+02 0.400000E+07 0.200000E+02 0.600000E+07
 0.200000E+02 0.800000E+07 0.200000E+02 0.100000E+08 0.200000E+02
 0.120000E+08 0.200000E+02 0.150000E+08 0.200000E+02 0.200000E+08
 0.200000E+02
THE DATA TYPE SEDIC     HAS 1 FURTHER NAMES AND   11 NAME COMBINATIONS :
 0.600000E+06 0.105000E+07 0.150000E+07 0.240000E+07 0.420000E+07
 0.600000E+07 0.780000E+07 0.114000E+08 0.150000E+08 0.175000E+08
 0.200000E+08
THE DATA TYPE SED2N     HAS 1 FURTHER NAMES AND    8 NAME COMBINATIONS :
 0.600000E+07 0.700000E+07 0.800000E+07 0.100000E+08 0.120000E+08
 0.140000E+08 0.150000E+08 0.200000E+08
THE DATA TYPE SED3N     HAS 1 FURTHER NAMES AND    5 NAME COMBINATIONS :
 0.132000E+08 0.135000E+08 0.140000E+08 0.150000E+08 0.200000E+08
THE DATA TYPE SGIZ      HAS 1 FURTHER NAMES AND    7 NAME COMBINATIONS :
 0.404000E+05 0.920000E+05 0.312000E+06 0.340000E+06 0.399000E+06
 0.416000E+06 0.461000E+06
THE DATA TYPE SGNC      HAS 1 FURTHER NAMES AND   43 NAME COMBINATIONS :
 0.100000E-04 0.100000E+01 0.100000E+03 0.500000E+04 0.100000E+05
 0.200000E+05 0.500000E+05 0.600000E+05 0.800000E+05 0.100000E+06
 0.120000E+06 0.140000E+06 0.160000E+06 0.180000E+06 0.200000E+06
 0.250000E+06 0.300000E+06 0.350000E+06 0.400000E+06 0.500000E+06
 0.610000E+06 0.650000E+06 0.700000E+06 0.800000E+06 0.900000E+06
 0.910000E+06 0.950000E+06 0.100000E+07 0.120000E+07 0.140000E+07
 0.160000E+07 0.180000E+07 0.200000E+07 0.250000E+07 0.300000E+07
 0.350000E+07 0.400000E+07 0.600000E+07 0.800000E+07 0.100000E+08
 0.120000E+08 0.150000E+08 0.200000E+08
```

```
FOR U 234EN4    24 DATA TYPES ARE AVAILABLE :
  ISOT1      ISOT2      LEGNC      MUEL       NUE        RANGRES    RES
  SEDF       SEDIC      SED2N      SED3N      SGA        SGF        SGG
  SGI        SGIZ       SGIZC      SGN        SGNC       SGT        SGTR
  SG2N       SG3N       ST
THE DATA TYPE LEGNC      HAS 2 FURTHER NAMES AND   83 NAME COMBINATIONS :
 0.100000E-04 0.100000E+01 0.500000E+04 0.100000E+01 0.600000E+04
 0.100000E+01 0.700000E+04 0.100000E+01 0.800000E+04 0.100000E+01
 0.900000E+04 0.100000E+01 0.100000E+05 0.100000E+01 0.120000E+05
 0.100000E+01 0.140000E+05 0.100000E+01 0.160000E+05 0.100000E+01
 0.180000E+05 0.100000E+01 0.200000E+05 0.100000E+01 0.220000E+05
 0.100000E+01 0.240000E+05 0.100000E+01 0.260000E+05 0.100000E+01
 0.280000E+05 0.100000E+01 0.300000E+05 0.100000E+01 0.350000E+05
 0.100000E+01 0.400000E+05 0.100000E+01 0.450000E+05 0.200000E+01
 0.500000E+05 0.200000E+01 0.550000E+05 0.200000E+01 0.600000E+05
 0.200000E+01 0.650000E+05 0.200000E+01 0.700000E+05 0.200000E+01
 0.750000E+05 0.200000E+01 0.800000E+05 0.200000E+01 0.850000E+05
 0.200000E+01 0.900000E+05 0.300000E+01 0.950000E+05 0.300000E+01
 0.100000E+06 0.300000E+01 0.120000E+06 0.300000E+01 0.140000E+06
 0.300000E+01 0.160000E+06 0.400000E+01 0.180000E+06 0.400000E+01
 0.200000E+06 0.400000E+01 0.220000E+06 0.400000E+01 0.240000E+06
 0.400000E+01 0.260000E+06 0.500000E+01 0.280000E+06 0.500000E+01
 0.300000E+06 0.500000E+01 0.350000E+06 0.600000E+01 0.400000E+06
 0.600000E+01 0.450000E+06 0.600000E+01 0.500000E+06 0.600000E+01
 0.550000E+06 0.600000E+01 0.600000E+06 0.600000E+01 0.650000E+06
 0.600000E+01 0.700000E+06 0.600000E+01 0.750000E+06 0.600000E+01
 0.800000E+06 0.600000E+01 0.850000E+06 0.600000E+01 0.900000E+06
 0.600000E+01 0.950000E+06 0.600000E+01 0.100000E+07 0.600000E+01
 0.120000E+07 0.600000E+01 0.140000E+07 0.600000E+01 0.160000E+07
 0.600000E+01 0.180000E+07 0.600000E+01 0.200000E+07 0.600000E+01
 0.220000E+07 0.600000E+01 0.240000E+07 0.700000E+01 0.260000E+07
 0.700000E+01 0.280000E+07 0.700000E+01 0.300000E+07 0.900000E+01
 0.350000E+07 0.100000E+02 0.400000E+07 0.100000E+02 0.450000E+07
 0.100000E+02 0.500000E+07 0.110000E+02 0.550000E+07 0.120000E+02
 0.600000E+07 0.120000E+02 0.650000E+07 0.130000E+02 0.700000E+07
 0.130000E+02 0.750000E+07 0.130000E+02 0.800000E+07 0.140000E+02
 0.900000E+07 0.140000E+02 0.100000E+08 0.140000E+02 0.110000E+08
 0.140000E+02 0.120000E+08 0.150000E+02 0.130000E+08 0.150000E+02
 0.140000E+08 0.150000E+02 0.150000E+08 0.150000E+02 0.200000E+08
 0.150000E+02
THE DATA TYPE SEDF       HAS 1 FURTHER NAMES AND   36 NAME COMBINATIONS :
 0.100000E-04 0.175622E-02 0.350245E-02 0.699490E-02 0.139798E-01
 0.279496E-01 0.558892E-01 0.111768E+00 0.223527E+00 0.447044E+00
 0.894078E+00 0.178815E+01 0.357629E+01 0.715256E+01 0.143051E+02
 0.286102E+02 0.572205E+02 0.114441E+03 0.228882E+03 0.457764E+03
 0.915527E+03 0.183105E+04 0.366211E+04 0.732422E+04 0.146484E+05
 0.292969E+05 0.585937E+05 0.117188E+06 0.234375E+06 0.468750E+06
 0.937500E+06 0.187500E+07 0.375000E+07 0.750000E+07 0.150000E+08
 0.200000E+08
THE DATA TYPE SEDIC      HAS 1 FURTHER NAMES AND    9 NAME COMBINATIONS :
 0.900000E+06 0.134063E+07 0.178125E+07 0.266250E+07 0.442500E+07
 0.795000E+07 0.114750E+08 0.150000E+08 0.200000E+08
THE DATA TYPE SED2N      HAS 1 FURTHER NAMES AND    9 NAME COMBINATIONS :
 0.681840E+07 0.784110E+07 0.886380E+07 0.988650E+07 0.109092E+08
 0.129546E+08 0.150000E+08 0.175000E+08 0.200000E+08
THE DATA TYPE SED3N      HAS 1 FURTHER NAMES AND   13 NAME COMBINATIONS :
 0.500000E+06 0.726563E+06 0.953125E+06 0.140625E+07 0.231250E+07
 0.321875E+07 0.412500E+07 0.593750E+07 0.775000E+07 0.113750E+08
 0.127300E+08 0.150000E+08 0.200000E+08
THE DATA TYPE SGIZ       HAS 1 FURTHER NAMES AND    6 NAME COMBINATIONS :
 0.440000E+05 0.144000E+06 0.297000E+06 0.800000E+06 0.945000E+06
 0.103500E+07
```

```
THE DATA TYPE SGNC      HAS 1 FURTHER NAMES AND    83 NAME COMBINATIONS :
 0.100000E-04 0.500000E+04 0.600000E+04 0.700000E+04 0.800000E+04
 0.900000E+04 0.100000E+05 0.120000E+05 0.140000E+05 0.160000E+05
 0.180000E+05 0.200000E+05 0.220000E+05 0.240000E+05 0.260000E+05
 0.280000E+05 0.300000E+05 0.350000E+05 0.400000E+05 0.450000E+05
 0.500000E+05 0.550000E+05 0.600000E+05 0.650000E+05 0.700000E+05
 0.750000E+05 0.800000E+05 0.850000E+05 0.900000E+05 0.950000E+05
 0.100000E+06 0.120000E+06 0.140000E+06 0.160000E+06 0.180000E+06
 0.200000E+06 0.220000E+06 0.240000E+06 0.260000E+06 0.280000E+06
 0.300000E+06 0.350000E+06 0.400000E+06 0.450000E+06 0.500000E+06
 0.550000E+06 0.600000E+06 0.650000E+06 0.700000E+06 0.750000E+06
 0.800000E+06 0.850000E+06 0.900000E+06 0.950000E+06 0.100000E+07
 0.120000E+07 0.140000E+07 0.160000E+07 0.180000E+07 0.200000E+07
 0.220000E+07 0.240000E+07 0.260000E+07 0.280000E+07 0.300000E+07
 0.350000E+07 0.400000E+07 0.450000E+07 0.500000E+07 0.550000E+07
 0.600000E+07 0.650000E+07 0.700000E+07 0.750000E+07 0.800000E+07
 0.900000E+07 0.100000E+08 0.110000E+08 0.120000E+08 0.130000E+08
 0.140000E+08 0.150000E+08 0.200000E+08


FOR U 236EN4     24 DATA TYPES ARE AVAILABLE :
 ISOT1     ISOT2     LEGNC     MUEL      NUE       RANGRES   RES
 SEDF      SEDIC     SED2N     SED3N     SGA       SGF       SGG
 SGI       SGIZ      SGIZC     SGN       SGNC      SGT       SGTR
 SG2N      SG3N      ST
THE DATA TYPE LEGNC     HAS 2 FURTHER NAMES AND    83 NAME COMBINATIONS :
 0.100000E-04 0.100000E+01 0.500000E+04 0.100000E+01 0.600000E+04
 0.100000E+01 0.700000E+04 0.100000E+01 0.800000E+04 0.100000E+01
 0.900000E+04 0.100000E+01 0.100000E+05 0.100000E+01 0.120000E+05
 0.100000E+01 0.140000E+05 0.100000E+01 0.160000E+05 0.100000E+01
 0.180000E+05 0.100000E+01 0.200000E+05 0.100000E+01 0.220000E+05
 0.100000E+01 0.240000E+05 0.100000E+01 0.260000E+05 0.100000E+01
 0.280000E+05 0.100000E+01 0.300000E+05 0.100000E+01 0.350000E+05
 0.100000E+01 0.400000E+05 0.100000E+01 0.450000E+05 0.200000E+01
 0.500000E+05 0.200000E+01 0.550000E+05 0.200000E+01 0.600000E+05
 0.200000E+01 0.650000E+05 0.200000E+01 0.700000E+05 0.200000E+01
 0.750000E+05 0.200000E+01 0.800000E+05 0.200000E+01 0.850000E+05
 0.200000E+01 0.900000E+05 0.300000E+01 0.950000E+05 0.300000E+01
 0.100000E+06 0.300000E+01 0.120000E+06 0.300000E+01 0.140000E+06
 0.300000E+01 0.160000E+06 0.400000E+01 0.180000E+06 0.400000E+01
 0.200000E+06 0.400000E+01 0.220000E+06 0.400000E+01 0.240000E+06
 0.400000E+01 0.260000E+06 0.500000E+01 0.280000E+06 0.500000E+01
 0.300000E+06 0.500000E+01 0.350000E+06 0.600000E+01 0.400000E+06
 0.600000E+01 0.450000E+06 0.600000E+01 0.500000E+06 0.600000E+01
 0.550000E+06 0.600000E+01 0.600000E+06 0.600000E+01 0.650000E+06
 0.600000E+01 0.700000E+06 0.600000E+01 0.750000E+06 0.600000E+01
 0.800000E+06 0.600000E+01 0.850000E+06 0.600000E+01 0.900000E+06
 0.600000E+01 0.950000E+06 0.600000E+01 0.100000E+07 0.600000E+01
 0.120000E+07 0.600000E+01 0.140000E+07 0.600000E+01 0.160000E+07
 0.600000E+01 0.180000E+07 0.600000E+01 0.200000E+07 0.600000E+01
 0.220000E+07 0.600000E+01 0.240000E+07 0.700000E+01 0.260000E+07
 0.700000E+01 0.280000E+07 0.700000E+01 0.300000E+07 0.900000E+01
 0.350000E+07 0.100000E+02 0.400000E+07 0.100000E+02 0.450000E+07
 0.100000E+02 0.500000E+07 0.110000E+02 0.550000E+07 0.120000E+02
 0.600000E+07 0.120000E+02 0.650000E+07 0.130000E+02 0.700000E+07
 0.130000E+02 0.750000E+07 0.130000E+02 0.800000E+07 0.140000E+02
 0.900000E+07 0.140000E+02 0.100000E+08 0.140000E+02 0.110000E+08
 0.140000E+02 0.120000E+08 0.150000E+02 0.130000E+08 0.150000E+02
 0.140000E+08 0.150000E+02 0.150000E+08 0.150000E+02 0.200000E+08
 0.150000E+02
```

```
THE DATA TYPE SEDF      HAS 1 FURTHER NAMES AND   36 NAME COMBINATIONS :
 0.100000E-04 0.175622E-02 0.350245E-02 0.699490E-02 0.139798E-01
 0.279496E-01 0.558892E-01 0.111768E+00 0.223527E+00 0.447044E+00
 0.894078E+00 0.178815E+01 0.357629E+01 0.715256E+01 0.143051E+02
 0.286102E+02 0.572205E+02 0.114441E+03 0.228882E+03 0.457764E+03
 0.915527E+03 0.183105E+04 0.366211E+04 0.732422E+04 0.146484E+05
 0.292969E+05 0.585937E+05 0.117188E+06 0.234375E+06 0.468750E+06
 0.937500E+06 0.187500E+07 0.375000E+07 0.750000E+07 0.150000E+08
 0.200000E+08
THE DATA TYPE SEDIC     HAS 1 FURTHER NAMES AND    3 NAME COMBINATIONS :
 0.900000E+06 0.150000E+08 0.200000E+08
THE DATA TYPE SED2N     HAS 1 FURTHER NAMES AND   13 NAME COMBINATIONS :
 0.500000E+06 0.726563E+06 0.953125E+06 0.140625E+07 0.231250E+07
 0.321875E+07 0.412500E+07 0.593750E+07 0.696800E+07 0.775000E+07
 0.113750E+08 0.150000E+08 0.200000E+08
THE DATA TYPE SED3N     HAS 1 FURTHER NAMES AND   13 NAME COMBINATIONS :
 0.500000E+06 0.726563E+06 0.953125E+06 0.140625E+07 0.231250E+07
 0.321875E+07 0.412500E+07 0.593750E+07 0.775000E+07 0.113750E+08
 0.117380E+08 0.150000E+08 0.200000E+08
THE DATA TYPE SGIZ      HAS 1 FURTHER NAMES AND    6 NAME COMBINATIONS :
 0.452800E+05 0.146000E+06 0.298000E+06 0.695000E+06 0.980000E+06
 0.106000E+07
THE DATA TYPE SGNC      HAS 1 FURTHER NAMES AND   83 NAME COMBINATIONS :
 0.100000E-04 0.500000E+04 0.600000E+04 0.700000E+04 0.800000E+04
 0.900000E+04 0.100000E+05 0.120000E+05 0.140000E+05 0.160000E+05
 0.180000E+05 0.200000E+05 0.220000E+05 0.240000E+05 0.260000E+05
 0.280000E+05 0.300000E+05 0.350000E+05 0.400000E+05 0.450000E+05
 0.500000E+05 0.550000E+05 0.600000E+05 0.650000E+05 0.700000E+05
 0.750000E+05 0.800000E+05 0.850000E+05 0.900000E+05 0.950000E+05
 0.100000E+06 0.120000E+06 0.140000E+06 0.160000E+06 0.180000E+06
 0.200000E+06 0.220000E+06 0.240000E+06 0.260000E+06 0.280000E+06
 0.300000E+06 0.350000E+06 0.400000E+06 0.450000E+06 0.500000E+06
 0.550000E+06 0.600000E+06 0.650000E+06 0.700000E+06 0.750000E+06
 0.800000E+06 0.850000E+06 0.900000E+06 0.950000E+06 0.100000E+07
 0.120000E+07 0.140000E+07 0.160000E+07 0.180000E+07 0.200000E+07
 0.220000E+07 0.240000E+07 0.260000E+07 0.280000E+07 0.300000E+07
 0.350000E+07 0.400000E+07 0.450000E+07 0.500000E+07 0.550000E+07
 0.600000E+07 0.650000E+07 0.700000E+07 0.750000E+07 0.800000E+07
 0.900000E+07 0.100000E+08 0.110000E+08 0.120000E+08 0.130000E+08
 0.140000E+08 0.150000E+08 0.200000E+08
```

```
FOR PA233EN4    25 DATA TYPES ARE AVAILABLE :
 ISOT1       ISOT2       LEGNC       MUEL        NUE         RANGRES     RES
 SEDF        SEDIC       SED2N       SED3N       SGA         SGF         SGG
 SGI         SGIZ        SGIZC       SGN         SGNC        SGT         SGTR
 SGX         SG2N        SG3N        ST
THE DATA TYPE LEGNC      HAS 2 FURTHER NAMES AND    34 NAME COMBINATIONS :
 0.100000E-04 0.100000E+01 0.500000E+04 0.100000E+01 0.700000E+04
 0.100000E+01 0.100000E+05 0.100000E+01 0.125000E+05 0.100000E+01
 0.150000E+05 0.100000E+01 0.200000E+05 0.100000E+01 0.250000E+05
 0.100000E+01 0.300000E+05 0.100000E+01 0.400000E+05 0.100000E+01
 0.500000E+05 0.200000E+01 0.700000E+05 0.200000E+01 0.100000E+06
 0.300000E+01 0.125000E+06 0.400000E+01 0.150000E+06 0.400000E+01
 0.200000E+06 0.400000E+01 0.250000E+06 0.400000E+01 0.300000E+06
 0.500000E+01 0.400000E+06 0.600000E+01 0.500000E+06 0.600000E+01
 0.700000E+06 0.600000E+01 0.100000E+07 0.600000E+01 0.125000E+07
 0.600000E+01 0.150000E+07 0.600000E+01 0.200000E+07 0.600000E+01
 0.250000E+07 0.700000E+01 0.300000E+07 0.800000E+01 0.400000E+07
 0.100000E+02 0.500000E+07 0.100000E+02 0.700000E+07 0.120000E+02
 0.100000E+08 0.140000E+02 0.125000E+08 0.150000E+02 0.150000E+08
 0.150000E+02 0.200000E+08 0.150000E+02
THE DATA TYPE SEDF       HAS 1 FURTHER NAMES AND     6 NAME COMBINATIONS :
 0.0          0.100000E-04 0.479995E+06 0.480000E+06 0.150000E+08
 0.200000E+08
THE DATA TYPE SEDIC      HAS 1 FURTHER NAMES AND    15 NAME COMBINATIONS :
 0.199998E+06 0.200000E+06 0.315625E+06 0.431250E+06 0.662500E+06
 0.112500E+07 0.158750E+07 0.205000E+07 0.297500E+07 0.390000E+07
 0.575000E+07 0.760000E+07 0.113000E+08 0.150000E+08 0.200000E+08
THE DATA TYPE SED2N      HAS 1 FURTHER NAMES AND     9 NAME COMBINATIONS :
 0.668440E+07 0.672440E+07 0.800000E+07 0.825000E+07 0.850000E+07
 0.875000E+07 0.900000E+07 0.150000E+08 0.200000E+08
THE DATA TYPE SED3N      HAS 1 FURTHER NAMES AND     4 NAME COMBINATIONS :
 0.122322E+08 0.122599E+08 0.150000E+08 0.200000E+08
THE DATA TYPE SGIZ       HAS 1 FURTHER NAMES AND     5 NAME COMBINATIONS :
 0.187000E+05 0.569000E+05 0.712000E+05 0.868000E+05 0.104000E+06
THE DATA TYPE SGNC       HAS 1 FURTHER NAMES AND    34 NAME COMBINATIONS :
 0.100000E-04 0.500000E+04 0.700000E+04 0.100000E+05 0.125000E+05
 0.150000E+05 0.200000E+05 0.250000E+05 0.300000E+05 0.400000E+05
 0.500000E+05 0.700000E+05 0.100000E+06 0.125000E+06 0.150000E+06
 0.200000E+06 0.250000E+06 0.300000E+06 0.400000E+06 0.500000E+06
 0.700000E+06 0.100000E+07 0.125000E+07 0.150000E+07 0.200000E+07
 0.250000E+07 0.300000E+07 0.400000E+07 0.500000E+07 0.700000E+07
 0.100000E+08 0.125000E+08 0.150000E+08 0.200000E+08
```

```
FOR NP237EN4    25 DATA TYPES ARE AVAILABLE :
 ISOT1        ISOT2        LEGNC        MUEL         NUE          RANGRES      RES
 SEDF         SEDIC        SED2N        SED3N        SGA          SGF          SGG
 SGI          SGIZ         SGIZC        SGN          SGNC         SGT          SGTR
 SG2N         SG3N         ST           STGF
THE DATA TYPE LEGNC      HAS 2 FURTHER NAMES AND    39 NAME COMBINATIONS :
 0.100000E-04 0.100000E+01 0.100000E+05 0.100000E+01 0.500000E+05
 0.900000E+01 0.100000E+06 0.600000E+01 0.207000E+06 0.900000E+01
 0.300000E+06 0.900000E+01 0.350000E+06 0.700000E+01 0.400000E+06
 0.900000E+01 0.450000E+06 0.900000E+01 0.470000E+06 0.900000E+01
 0.500000E+06 0.900000E+01 0.550000E+06 0.800000E+01 0.570000E+06
 0.900000E+01 0.600000E+06 0.900000E+01 0.650000E+06 0.900000E+01
 0.700000E+06 0.900000E+01 0.750000E+06 0.900000E+01 0.775000E+06
 0.900000E+01 0.800000E+06 0.900000E+01 0.850000E+06 0.900000E+01
 0.900000E+06 0.900000E+01 0.950000E+06 0.900000E+01 0.100000E+07
 0.400000E+01 0.105000E+07 0.900000E+01 0.110000E+07 0.900000E+01
 0.115000E+07 0.900000E+01 0.120000E+07 0.900000E+01 0.125000E+07
 0.900000E+01 0.129500E+07 0.900000E+01 0.135000E+07 0.900000E+01
 0.140000E+07 0.900000E+01 0.145000E+07 0.900000E+01 0.149500E+07
 0.900000E+01 0.200000E+07 0.600000E+01 0.250000E+07 0.700000E+01
 0.410000E+07 0.900000E+01 0.700000E+07 0.110000E+02 0.150000E+08
 0.200000E+02 0.200000E+08 0.200000E+02
THE DATA TYPE SEDF       HAS 1 FURTHER NAMES AND     2 NAME COMBINATIONS :
 0.100000E-04 0.200000E+08
THE DATA TYPE SEDIC      HAS 1 FURTHER NAMES AND    14 NAME COMBINATIONS :
 0.333400E+06 0.500000E+06 0.100000E+07 0.200000E+07 0.300000E+07
 0.400000E+07 0.500000E+07 0.600000E+07 0.700000E+07 0.800000E+07
 0.900000E+07 0.100000E+08 0.150000E+08 0.200000E+08
THE DATA TYPE SED2N      HAS 1 FURTHER NAMES AND     6 NAME COMBINATIONS :
 0.678900E+07 0.800000E+07 0.840000E+07 0.880000E+07 0.100000E+08
 0.200000E+08
THE DATA TYPE SED3N      HAS 1 FURTHER NAMES AND    27 NAME COMBINATIONS :
 0.123400E+08 0.124437E+08 0.125475E+08 0.126512E+08 0.127550E+08
 0.128587E+08 0.129625E+08 0.130662E+08 0.131700E+08 0.132737E+08
 0.133775E+08 0.134812E+08 0.135850E+08 0.136887E+08 0.137925E+08
 0.138962E+08 0.140000E+08 0.145000E+08 0.150000E+08 0.156250E+08
 0.162500E+08 0.168750E+08 0.175000E+08 0.181250E+08 0.187500E+08
 0.193750E+08 0.200000E+08
THE DATA TYPE SGIZ       HAS 1 FURTHER NAMES AND    11 NAME COMBINATIONS :
 0.332000E+05 0.596000E+05 0.760000E+05 0.103000E+06 0.159000E+06
 0.224000E+06 0.268000E+06 0.305000E+06 0.332000E+06 0.369000E+06
 0.371000E+06
THE DATA TYPE SGNC       HAS 1 FURTHER NAMES AND    39 NAME COMBINATIONS :
 0.100000E-04 0.100000E+05 0.500000E+05 0.100000E+06 0.207000E+06
 0.300000E+06 0.350000E+06 0.400000E+06 0.450000E+06 0.470000E+06
 0.500000E+06 0.550000E+06 0.570000E+06 0.600000E+06 0.650000E+06
 0.700000E+06 0.750000E+06 0.775000E+06 0.800000E+06 0.850000E+06
 0.900000E+06 0.950000E+06 0.100000E+07 0.105000E+07 0.110000E+07
 0.115000E+07 0.120000E+07 0.125000E+07 0.129500E+07 0.135000E+07
 0.140000E+07 0.145000E+07 0.149500E+07 0.200000E+07 0.250000E+07
 0.410000E+07 0.700000E+07 0.150000E+08 0.200000E+08
```

```
FOR AM243EN4     19 DATA TYPES ARE AVAILABLE :
 ISOT1       ISOT2       LEGNC       MUEL        NUE         RANGRES    RES
 SEDF        SEDIC       SGA         SGF         SGG         SGI        SGIZC
 SGN         SGNC        SGT         SGTR        ST
THE DATA TYPE LEGNC      HAS 2 FURTHER NAMES AND    39 NAME COMBINATIONS :
 0.100000E-04 0.100000E+01 0.100000E+05 0.100000E+01 0.500000E+05
 0.900000E+01 0.100000E+06 0.600000E+01 0.207000E+06 0.900000E+01
 0.300000E+06 0.900000E+01 0.350000E+06 0.700000E+01 0.400000E+06
 0.900000E+01 0.450000E+06 0.900000E+01 0.470000E+06 0.900000E+01
 0.500000E+06 0.900000E+01 0.550000E+06 0.800000E+01 0.570000E+06
 0.900000E+01 0.600000E+06 0.900000E+01 0.650000E+06 0.900000E+01
 0.700000E+06 0.900000E+01 0.750000E+06 0.900000E+01 0.775000E+06
 0.900000E+01 0.800000E+06 0.900000E+01 0.850000E+06 0.900000E+01
 0.900000E+06 0.900000E+01 0.950000E+06 0.900000E+01 0.100000E+07
 0.400000E+01 0.105000E+07 0.900000E+01 0.110000E+07 0.900000E+01
 0.115000E+07 0.900000E+01 0.120000E+07 0.900000E+01 0.125000E+07
 0.900000E+01 0.129500E+07 0.900000E+01 0.135000E+07 0.900000E+01
 0.140000E+07 0.900000E+01 0.145000E+07 0.900000E+01 0.149500E+07
 0.900000E+01 0.200000E+07 0.600000E+01 0.250000E+07 0.700000E+01
 0.410000E+07 0.900000E+01 0.700000E+07 0.110000E+02 0.150000E+08
 0.200000E+02 0.200000E+08 0.200000E+02
THE DATA TYPE SEDF       HAS 1 FURTHER NAMES AND     3 NAME COMBINATIONS :
 0.100000E-04 0.150000E+08 0.200000E+08
THE DATA TYPE SEDIC      HAS 1 FURTHER NAMES AND    24 NAME COMBINATIONS :
 0.840000E+05 0.103750E+06 0.123500E+06 0.143250E+06 0.163000E+06
 0.202500E+06 0.242000E+06 0.281500E+06 0.321000E+06 0.400000E+06
 0.600000E+06 0.800000E+06 0.120000E+07 0.200000E+07 0.300000E+07
 0.400000E+07 0.500000E+07 0.600000E+07 0.700000E+07 0.800000E+07
 0.900000E+07 0.100000E+08 0.150000E+08 0.200000E+08
THE DATA TYPE SGNC       HAS 1 FURTHER NAMES AND    39 NAME COMBINATIONS :
 0.100000E-04 0.100000E+05 0.500000E+05 0.100000E+06 0.207000E+06
 0.300000E+06 0.350000E+06 0.400000E+06 0.450000E+06 0.470000E+06
 0.500000E+06 0.550000E+06 0.570000E+06 0.600000E+06 0.650000E+06
 0.700000E+06 0.750000E+06 0.775000E+06 0.800000E+06 0.850000E+06
 0.900000E+06 0.950000E+06 0.100000E+07 0.105000E+07 0.110000E+07
 0.115000E+07 0.120000E+07 0.125000E+07 0.129500E+07 0.135000E+07
 0.140000E+07 0.145000E+07 0.149500E+07 0.200000E+07 0.250000E+07
 0.410000E+07 0.700000E+07 0.150000E+08 0.200000E+08
```

```
FOR CM244EN4     25 DATA TYPES ARE AVAILABLE :
 ISOT1       ISOT2       LEGNC       MUEL        NUE         RANGRES     RES
 SEDF        SEDIC       SED2N       SED3N       SGA         SGF         SGG
 SGI         SGIZ        SGIZC       SGN         SGNC        SGT         SGTR
 SG2N        SG3N        ST          STGF
THE DATA TYPE LEGNC      HAS 2 FURTHER NAMES AND   39 NAME COMBINATIONS :
 0.100000E-04 0.100000E+01 0.100000E+05 0.100000E+01 0.200000E+05
 0.150000E+02 0.300000E+05 0.130000E+02 0.550000E+05 0.140000E+02
 0.650000E+05 0.150000E+02 0.750000E+05 0.600000E+01 0.900000E+05
 0.150000E+02 0.100000E+06 0.150000E+02 0.150000E+06 0.110000E+02
 0.200000E+06 0.100000E+02 0.300000E+06 0.150000E+02 0.400000E+06
 0.120000E+02 0.500000E+06 0.120000E+02 0.600000E+06 0.130000E+02
 0.700000E+06 0.140000E+02 0.800000E+06 0.150000E+02 0.900000E+06
 0.150000E+02 0.100000E+07 0.150000E+02 0.115000E+07 0.150000E+02
 0.125000E+07 0.110000E+02 0.150000E+07 0.110000E+02 0.175000E+07
 0.140000E+02 0.200000E+07 0.130000E+02 0.250000E+07 0.150000E+02
 0.300000E+07 0.150000E+02 0.400000E+07 0.150000E+02 0.500000E+07
 0.150000E+02 0.600000E+07 0.150000E+02 0.700000E+07 0.150000E+02
 0.800000E+07 0.150000E+02 0.900000E+07 0.150000E+02 0.100000E+08
 0.150000E+02 0.110000E+08 0.150000E+02 0.120000E+08 0.190000E+02
 0.130000E+08 0.190000E+02 0.140000E+08 0.190000E+02 0.150000E+08
 0.190000E+02 0.200000E+08 0.190000E+02
THE DATA TYPE SEDF       HAS 1 FURTHER NAMES AND    3 NAME COMBINATIONS :
 0.100000E-04 0.150000E+08 0.200000E+08
THE DATA TYPE SEDIC      HAS 1 FURTHER NAMES AND   19 NAME COMBINATIONS :
 0.100000E+07 0.150000E+07 0.170000E+07 0.200000E+07 0.250000E+07
 0.300000E+07 0.400000E+07 0.500000E+07 0.600000E+07 0.700000E+07
 0.800000E+07 0.900000E+07 0.100000E+08 0.110000E+08 0.120000E+08
 0.130000E+08 0.140000E+08 0.150000E+08 0.200000E+08
THE DATA TYPE SED2N      HAS 1 FURTHER NAMES AND   12 NAME COMBINATIONS :
 0.670000E+07 0.700000E+07 0.750000E+07 0.800000E+07 0.900000E+07
 0.100000E+08 0.110000E+08 0.120000E+08 0.130000E+08 0.140000E+08
 0.150000E+08 0.200000E+08
THE DATA TYPE SED3N      HAS 1 FURTHER NAMES AND    5 NAME COMBINATIONS :
 0.124000E+08 0.130000E+08 0.140000E+08 0.150000E+08 0.200000E+08
THE DATA TYPE SGIZ       HAS 1 FURTHER NAMES AND   10 NAME COMBINATIONS :
 0.429000E+05 0.142300E+06 0.296000E+06 0.598000E+06 0.658000E+06
 0.743000E+06 0.920000E+06 0.979000E+06 0.101000E+07 0.105000E+07
THE DATA TYPE SGNC       HAS 1 FURTHER NAMES AND   39 NAME COMBINATIONS :
 0.100000E-04 0.100000E+05 0.200000E+05 0.300000E+05 0.550000E+05
 0.650000E+05 0.750000E+05 0.900000E+05 0.100000E+06 0.150000E+06
 0.200000E+06 0.300000E+06 0.400000E+06 0.500000E+06 0.600000E+06
 0.700000E+06 0.800000E+06 0.900000E+06 0.100000E+07 0.115000E+07
 0.125000E+07 0.150000E+07 0.175000E+07 0.200000E+07 0.250000E+07
 0.300000E+07 0.400000E+07 0.500000E+07 0.600000E+07 0.700000E+07
 0.800000E+07 0.900000E+07 0.100000E+08 0.110000E+08 0.120000E+08
 0.130000E+08 0.140000E+08 0.150000E+08 0.200000E+08



FOR TA181EN4     18 DATA TYPES ARE AVAILABLE :
 ISOT1       ISOT2       MUEL        RANGRES     RES         SGA         SGG
 SGI         SGIZ        SGIZC       SGN         SGNC        SGP         SGT
 SGTR        SG2N        SG3N        ST
THE DATA TYPE SGIZ       HAS 1 FURTHER NAMES AND   10 NAME COMBINATIONS :
 0.620000E+04 0.136100E+06 0.158600E+06 0.301500E+06 0.337500E+06
 0.482200E+06 0.495000E+06 0.620000E+06 0.720000E+06 0.925000E+06
THE DATA TYPE SGNC       HAS 1 FURTHER NAMES AND   19 NAME COMBINATIONS :
 0.100000E-04 0.500000E+05 0.750000E+05 0.130000E+06 0.330000E+06
 0.560000E+06 0.750000E+06 0.100000E+07 0.200000E+07 0.400000E+07
 0.600000E+07 0.800000E+07 0.100000E+08 0.120000E+08 0.140000E+08
 0.150000E+08 0.160000E+08 0.180000E+08 0.200000E+08
```

```
FOR NB 93EN4     21 DATA TYPES ARE AVAILABLE :
 ISOT1       ISOT2       LEGNC       MUEL        RANGRES     RES         SGA
 SGALP       SGG         SGI         SGIA        SGIZ        SGIZC       SGN
 SGNC        SGP         SGT         SGTR        SG2N        SG3N        ST
THE DATA TYPE LEGNC      HAS 2 FURTHER NAMES AND   33 NAME COMBINATIONS :
 0.100000E-04 0.100000E+01 0.100000E+05 0.100000E+01 0.100000E+06
 0.200000E+01 0.200000E+06 0.200000E+01 0.300000E+06 0.400000E+01
 0.400000E+06 0.400000E+01 0.500000E+06 0.400000E+01 0.600000E+06
 0.400000E+01 0.700000E+06 0.400000E+01 0.800000E+06 0.400000E+01
 0.900000E+06 0.400000E+01 0.100000E+07 0.400000E+01 0.120000E+07
 0.400000E+01 0.140000E+07 0.600000E+01 0.160000E+07 0.600000E+01
 0.180000E+07 0.600000E+01 0.200000E+07 0.600000E+01 0.250000E+07
 0.600000E+01 0.300000E+07 0.600000E+01 0.350000E+07 0.800000E+01
 0.400000E+07 0.800000E+01 0.450000E+07 0.800000E+01 0.500000E+07
 0.800000E+01 0.600000E+07 0.100000E+02 0.700000E+07 0.100000E+02
 0.800000E+07 0.100000E+02 0.900000E+07 0.120000E+02 0.100000E+08
 0.120000E+02 0.120000E+08 0.120000E+02 0.140000E+08 0.120000E+02
 0.160000E+08 0.140000E+02 0.180000E+08 0.140000E+02 0.200000E+08
 0.140000E+02
THE DATA TYPE SGIZ       HAS 1 FURTHER NAMES AND   12 NAME COMBINATIONS :
 0.290000E+05 0.740000E+06 0.810000E+06 0.959000E+06 0.107000E+07
 0.131500E+07 0.148840E+07 0.167400E+07 0.194700E+07 0.215900E+07
 0.233500E+07 0.251900E+07
THE DATA TYPE SGNC       HAS 1 FURTHER NAMES AND   33 NAME COMBINATIONS :
 0.100000E-04 0.100000E+05 0.100000E+06 0.200000E+06 0.300000E+06
 0.400000E+06 0.500000E+06 0.600000E+06 0.700000E+06 0.800000E+06
 0.900000E+06 0.100000E+07 0.120000E+07 0.140000E+07 0.160000E+07
 0.180000E+07 0.200000E+07 0.250000E+07 0.300000E+07 0.350000E+07
 0.400000E+07 0.450000E+07 0.500000E+07 0.600000E+07 0.700000E+07
 0.800000E+07 0.900000E+07 0.100000E+08 0.120000E+08 0.140000E+08
 0.160000E+08 0.180000E+08 0.200000E+08
```

```
FOR NB 93RCN     16 DATA TYPES ARE AVAILABLE :
 AASTATUS    ISOT1       ISOT2       MUEL        RANGRES     RES         SGA
 SGG         SGI         SGIZ        SGIZC       SGN         SGT         SG2N
 ST          STD
THE DATA TYPE SGIZ       HAS 1 FURTHER NAMES AND    9 NAME COMBINATIONS :
 0.304000E+05 0.686000E+06 0.744000E+06 0.808700E+06 0.810100E+06
 0.949900E+06 0.979100E+06 0.108300E+07 0.112700E+07
```

```
FOR MN 55EN4     26 DATA TYPES ARE AVAILABLE :
 ISOT1       ISOT2       LEGNC       MUEL        RANGRES     RES         SEDIC
 SED2N       SED3N       SGA         SGALP       SGD         SGG         SGHE3
 SGI         SGIA        SGIP        SGIZ        SGIZC       SGN         SGNC
 SGP         SGT         SGTR        SG2N        SG3N
THE DATA TYPE LEGNC      HAS 2 FURTHER NAMES AND   86 NAME COMBINATIONS :
 0.100000E-04 0.100000E+01 0.100000E+05 0.100000E+01 0.677000E+06
 0.400000E+01 0.686000E+06 0.400000E+01 0.694000E+06 0.400000E+01
 0.703000E+06 0.400000E+01 0.711000E+06 0.400000E+01 0.719000E+06
 0.400000E+01 0.728000E+06 0.400000E+01 0.736000E+06 0.400000E+01
 0.745000E+06 0.400000E+01 0.753000E+06 0.400000E+01 0.761000E+06
 0.400000E+01 0.770000E+06 0.400000E+01 0.778000E+06 0.400000E+01
 0.787000E+06 0.400000E+01 0.795000E+06 0.400000E+01 0.803000E+06
 0.400000E+01 0.812000E+06 0.400000E+01 0.820000E+06 0.500000E+01
 0.829000E+06 0.400000E+01 0.837000E+06 0.400000E+01 0.854000E+06
 0.400000E+01 0.871000E+06 0.400000E+01 0.879000E+06 0.400000E+01
 0.887000E+06 0.400000E+01 0.896000E+06 0.400000E+01 0.904000E+06
 0.400000E+01 0.913000E+06 0.400000E+01 0.921000E+06 0.400000E+01
 0.929000E+06 0.400000E+01 0.938000E+06 0.400000E+01 0.946000E+06
 0.400000E+01 0.955000E+06 0.500000E+01 0.963000E+06 0.400000E+01
 0.971000E+06 0.400000E+01 0.980000E+06 0.400000E+01 0.988000E+06
 0.400000E+01 0.997000E+06 0.400000E+01 0.100500E+07 0.400000E+01
 0.101300E+07 0.400000E+01 0.102200E+07 0.400000E+01 0.103000E+07
 0.400000E+01 0.103900E+07 0.400000E+01 0.104700E+07 0.400000E+01
 0.105500E+07 0.400000E+01 0.106400E+07 0.400000E+01 0.107200E+07
 0.400000E+01 0.108100E+07 0.400000E+01 0.108700E+07 0.500000E+01
 0.108900E+07 0.400000E+01 0.109500E+07 0.400000E+01 0.110400E+07
 0.400000E+01 0.111200E+07 0.500000E+01 0.112000E+07 0.500000E+01
 0.112900E+07 0.400000E+01 0.113700E+07 0.400000E+01 0.114500E+07
 0.500000E+01 0.115400E+07 0.500000E+01 0.116200E+07 0.400000E+01
 0.117000E+07 0.400000E+01 0.117900E+07 0.500000E+01 0.118700E+07
 0.400000E+01 0.119600E+07 0.500000E+01 0.120400E+07 0.500000E+01
 0.121200E+07 0.500000E+01 0.122100E+07 0.500000E+01 0.247000E+07
 0.700000E+01 0.300000E+07 0.600000E+01 0.349000E+07 0.700000E+01
 0.400000E+07 0.400000E+01 0.456000E+07 0.900000E+01 0.609000E+07
 0.800000E+01 0.705000E+07 0.900000E+01 0.805000E+07 0.900000E+01
 0.900000E+07 0.190000E+02 0.110000E+08 0.190000E+02 0.120000E+08
 0.190000E+02 0.130000E+08 0.190000E+02 0.140000E+08 0.190000E+02
 0.150000E+08 0.190000E+02 0.160000E+08 0.190000E+02 0.170000E+08
 0.190000E+02 0.180000E+08 0.190000E+02 0.190000E+08 0.190000E+02
 0.200000E+08 0.190000E+02
THE DATA TYPE SEDIC     HAS 1 FURTHER NAMES AND   15 NAME COMBINATIONS :
 0.191768E+07 0.700000E+07 0.800000E+07 0.900000E+07 0.100000E+08
 0.110000E+08 0.120000E+08 0.130000E+08 0.140000E+08 0.150000E+08
 0.160000E+08 0.170000E+08 0.180000E+08 0.190000E+08 0.200000E+08
THE DATA TYPE SED2N     HAS 1 FURTHER NAMES AND    8 NAME COMBINATIONS :
 0.104127E+08 0.110000E+08 0.150000E+08 0.160000E+08 0.170000E+08
 0.180000E+08 0.190000E+08 0.200000E+08
THE DATA TYPE SED3N     HAS 1 FURTHER NAMES AND    2 NAME COMBINATIONS :
 0.195199E+08 0.200000E+08
THE DATA TYPE SGIZ      HAS 1 FURTHER NAMES AND    5 NAME COMBINATIONS :
 0.125800E+06 0.984000E+06 0.129200E+07 0.152800E+07 0.188300E+07
THE DATA TYPE SGNC      HAS 1 FURTHER NAMES AND   86 NAME COMBINATIONS :
 0.100000E-04 0.100000E+05 0.677000E+06 0.686000E+06 0.694000E+06
 0.703000E+06 0.711000E+06 0.719000E+06 0.728000E+06 0.736000E+06
 0.745000E+06 0.753000E+06 0.761000E+06 0.770000E+06 0.778000E+06
 0.787000E+06 0.795000E+06 0.803000E+06 0.812000E+06 0.820000E+06
 0.829000E+06 0.837000E+06 0.854000E+06 0.871000E+06 0.879000E+06
 0.887000E+06 0.896000E+06 0.904000E+06 0.913000E+06 0.921000E+06
 0.929000E+06 0.938000E+06 0.946000E+06 0.955000E+06 0.963000E+06
 0.971000E+06 0.980000E+06 0.988000E+06 0.997000E+06 0.100500E+07
 0.101300E+07 0.102200E+07 0.103000E+07 0.103900E+07 0.104700E+07
 0.105500E+07 0.106400E+07 0.107200E+07 0.108100E+07 0.108700E+07
 0.108900E+07 0.109500E+07 0.110400E+07 0.111200E+07 0.112000E+07
 0.112900E+07 0.113700E+07 0.114500E+07 0.115400E+07 0.116200E+07
 0.117000E+07 0.117900E+07 0.118700E+07 0.119600E+07 0.120400E+07
 0.121200E+07 0.122100E+07 0.247000E+07 0.300000E+07 0.349000E+07
 0.400000E+07 0.456000E+07 0.609000E+07 0.705000E+07 0.805000E+07
 0.900000E+07 0.110000E+08 0.120000E+08 0.130000E+08 0.140000E+08
 0.150000E+08 0.160000E+08 0.170000E+08 0.180000E+08 0.190000E+08
 0.200000E+08
```

```
FOR P   31ENL     15 DATA TYPES ARE AVAILABLE :
 ISOT1        ISOT2        MUEL         SGA          SGALP        SGG          SGI
 SGIP         SGIZC        SGN          SGNC         SGP          SGT          SGTR
 SG2N
THE DATA TYPE SGNC       HAS 1 FURTHER NAMES AND    21 NAME COMBINATIONS :
 0.100000E-03 0.100000E+06 0.150000E+06 0.330000E+06 0.400000E+06
 0.570000E+06 0.980000E+06 0.114000E+07 0.138000E+07 0.178000E+07
 0.350000E+07 0.420000E+07 0.450000E+07 0.480000E+07 0.600000E+07
 0.100000E+08 0.120000E+08 0.146000E+08 0.160000E+08 0.180000E+08
 0.200000E+08


FOR GA    ENL     14 DATA TYPES ARE AVAILABLE :
 ISOT1        ISOT2        MUEL         SGA          SGALP        SGG          SGI
 SGIZC        SGN          SGNC         SGP          SGT          SGTR         SG2N
THE DATA TYPE SGNC       HAS 1 FURTHER NAMES AND    10 NAME COMBINATIONS :
 0.100000E-03 0.100000E+06 0.100000E+07 0.300000E+07 0.700000E+07
 0.110000E+08 0.146000E+08 0.160000E+08 0.180000E+08 0.200000E+08


FOR ZR    ENL     15 DATA TYPES ARE AVAILABLE :
 ISOT1        ISOT2        MUEL         SGA          SGG          SGI          SGIZ
 SGIZC        SGN          SGNC         SGP          SGT          SGTR         SG2N
 SG3N
THE DATA TYPE SGIZ       HAS 1 FURTHER NAMES AND     8 NAME COMBINATIONS :
 0.919800E+06 0.121000E+07 0.130000E+07 0.138000E+07 0.148000E+07
 0.166000E+07 0.175000E+07 0.185940E+07
THE DATA TYPE SGNC       HAS 1 FURTHER NAMES AND    21 NAME COMBINATIONS :
 0.100000E-03 0.100000E+05 0.600000E+05 0.140000E+06 0.230000E+06
 0.340000E+06 0.550000E+06 0.104000E+07 0.145000E+07 0.250000E+07
 0.290000E+07 0.410000E+07 0.470000E+07 0.700000E+07 0.800000E+07
 0.100000E+08 0.120000E+08 0.140000E+08 0.160000E+08 0.180000E+08
 0.200000E+08


FOR EU    ENL     13 DATA TYPES ARE AVAILABLE :
 ISOT1        ISOT2        MUEL         SGA          SGG          SGI          SGIZC
 SGN          SGNC         SGT          SGTR         SG2N         SG3N
THE DATA TYPE SGNC       HAS 1 FURTHER NAMES AND     9 NAME COMBINATIONS :
 0.100000E-03 0.500000E+06 0.100000E+07 0.400000E+07 0.700000E+07
 0.146000E+08 0.160000E+08 0.180000E+08 0.200000E+08


FOR W     ENL     13 DATA TYPES ARE AVAILABLE :
 ISOT1        ISOT2        MUEL         SGA          SGG          SGI          SGIZC
 SGN          SGNC         SGT          SGTR         SG2N         SG3N
THE DATA TYPE SGNC       HAS 1 FURTHER NAMES AND    12 NAME COMBINATIONS :
 0.100000E-03 0.500000E+05 0.100000E+06 0.325000E+06 0.495000E+06
 0.975000E+06 0.201000E+07 0.500000E+07 0.800000E+07 0.110000E+08
 0.140000E+08 0.200000E+08
```

Appendix 2: Lay-out of logical records on the KEDAK-file

```
        SUMMARY OF THE KARLSRUHE NUCLEAR DATA FILE

COMMENT  1

NEUTRON CROSS SECTIONS AND RELATED DATA

THE LAY-CUT OF DATA-RECORDS IS AS FOLLOWS
1ST NAME IS THE NAME OF THE ISOTOPE
2ND NAME IS THE NAME OF THE CROSS SECTION
3RD NAME IS THE ENERGY (GIVEN IN MEV) OF THE RESIDUAL NUCLEUS
    LEVEL FOR INELASTIC EXCITATION (FOR TYPE SGIZ ONLY)
THE ARGUMENT IS THE VALUE OF ENERGY. THIS IS GIVEN IN EV
THE DATAWORD IS THE VALUE OF THE CROSS SECTION. THIS IS GIVEN
    IN BARN

THE FOLLOWING TABLE SHOWS THE MEANING OF THE NAMES OF THE
    CROSS SECTIONS

    SGN     ELASTIC SCATTERING
    SGI     TOTAL INELASTIC SCATTERING
    SGIZ    INELASTIC SCATTERING OF LEVEL DEFINED BY 3RD NAME
    SGIZC   CONTINUUM PART OF THE INELASTIC SCATTERING
    SGG     RADIATIVE CAPTURE
    SGF     FISSION
    SG2N    (N,2N) PROCESS
    SG2NA   (N,2ALPHA) PROCESS
    SG3N    (N,3N) PROCESS
    SG2HE   (N,2ALPHA) PROCESS
    SGHE3   (N,HE3) PROCESS
    SGH3    (N,H3) PROCESS
    SGD     (N,D) PROCESS
    SGALP   (N,ALPHA) PROCESS
    SGP     (N,P) PROCESS
    SGT     TOTAL
    SGTR    TRANSPORT
    SGA     ABSORPTION          SGA=SGG+SGF+SGP+SGALP+SGD
                                      +SGH3+SGHE3+SG2HE
    SGX     NON ELASTIC         SGX=SGT - SGN
    MUEL    AVERAGE COSINE OF ELASTIC SCATTERING ANGLE
            IN LABOR SYSTEM
    CHIF    ENERGY DISTRIBUTION OF PROMPT FISSION NEUTRONS
    NUE     MEAN NUMBER OF SECONDARY NEUTRONS EMITTED PER
            FISSION
    ETA     EFFECTIVE NUMBER OF SECONDARY NEUTRONS EMITTED
            PER NEJTRON ABSORPTION
    ALPHA   SGG / SGF

WHERE APPLICABLE THE FOLLOWING RELATIONS BETWEEN CROSS
SECTIONS HAVE BEEN CHECKED



    SGT    = SGN + SGG + SGF + SGI + SGP + SGALP + SG2N + SGD
    SGI    = SUM(SGIZ) + SGIZC
    SGA    = SGG + SGF + SGP + SGALP + SGD
                 +SGH3 + SGHE3 + SG2HE
    SGTR   = SGT - MUEL * SGN
    SGX    = SGT - SGN
    ALPHA  = SGG / SGF
    ETA    = NUE / ( 1.+ ALPHA )
    INTEGRAL OVER CHIF(E) = 1.
```

```
COMMENT  2

1-6-70
AVERAGE ENERGY INDEPENDENT STATISTICAL THEORY PARAMETERS

THE LAY-OUT OF THE DATA-RECORD IS AS FOLLOWS
1ST NAME IS THE NAME OF THE ISOTOPE
2ND NAME IS STD
1ST DATAWORD IS THE OBSERVED AVERAGE LEVEL DISTANCE
2ND DATAWORD IS THE PARAMETER A OF THE STATISTICAL THEORY
3RD DATAWORD IS THE PARAMETER 2*SIGMA**2 OF THE STAT. THEORY


COMMENT  3

1-6-70
AVERAGE ENERGY DEPENDENT RESONANCE PARAMETERS
DRESNER FACTORS COMPUTED BY KARLSRUHE PROGRAM 01741

THE LAY-OUT OF THE DATA-RECORD IS AS FOLLOWS
1ST NAME IS THE NAME OF THE ISOTOPE
2ND NAME IS STGF
1ST ARGUMENT IS THE ENERGY
2ND ARGUMENT IS THE NEUTRON ORBITAL ANGULAR MOMENTUM L
3RD ARGUMENT IS THE TOTAL ANGULAR MOMENTUM OF THE COMPOUND
    NUCLEUS J
1ST DATAWORD IS NUE , THE DEGREE OF FREEDOM FOR THE FISSION
    WIDTH DISTRIBUTION
2ND DATAWORD IS THE AVERAGE FISSION WIDTH
3RD DATAWORD IS THE AVERAGE RADIATION WIDTH
4TH DATAWORD IS THE AVERAGE NEUTRON WIDTH
5TH , 6TH , 7TH , 8TH DATAWORDS ARE THE DRESNERFACTORS
    SF , SG , RF , RG


COMMENT  4

1-6-70
CHARACTERISTIC ISOTOPE DATA

THE LAY-OUT OF THE DATA-RECORD IS AS FOLLOWS
1ST NAME IS THE NAME OF THE ISOTOPE
2ND NAME IS ISOT1
1ST DATAWORD IS THE ATOMIC WEIGHT
2ND DATAWORD IS THE ATOMIC NUMBER
3RD DATAWORD IS THE GROUND-STATE SPIN I
```

```
COMMENT  5

1-6-70
CHARACTERISTIC ISOTOPE DATA

THE LAY-OUT OF THE DATA-RECORD IS AS FOLLOWS
1ST NAME IS THE NAME OF THE ISOTOPE
2ND NAME IS ISOT2
1ST DATAWORD IS THE ENERGY INDEPENDENT REDUCED NEUTRON
    WAVE LENGTH
2ND DATAWORD IS THE NUCLEAR RADIUS
3RD DATAWORD IS THE EFFECTIVE BINDING ENERGY OF THE LAST
    NEUTRON IN COMPOUND NUCLEUS


COMMENT  6

1-6-70
AVERAGE ENERGY INDEPENDENT RESONANCE PARAMETERS

THE LAY-OUT OF THE DATA-RECORD IS AS FOLLOWS
1ST NAME IS THE NAME OF THE ISOTOPE
2ND NAME IS ST
1ST ARGUMENT IS THE NEUTRON ORBITAL ANGULAR MOMENTUM L
2ND ARGUMENT IS THE TOTAL ANGULAR MOMENTUM OF THE COMPOUND
    NUCLEUS J
1ST DATAWORD IS THE AVERAGE RADIATION WIDTH
2ND DATAWORD IS THE AVERAGE LEVEL DISTANCE
3RD DATAWORD IS THE AVERAGE REDUCED NEUTRON WIDTH
4TH DATAWORD IS THE STRENGTH FUNCTION
5TH DATAWORD IS THE NUMBER OF FISSION CHANNELS
6TH DATAWORD IS THE NUMBER OF NEUTRON CHANNELS


COMMENT  7

1-6-70
RESOLVED RESONANCE PARAMETERS

THE LAY-OUT OF THE DATA-RECORD IS AS FOLLOWS
(ALL ENERGIES ARE GIVEN IN EV)
1ST NAME IS THE NAME OF THE ISOTOPE
2ND NAME IS RES
1ST ARGUMENT IS THE ENERGY OF RESONANCE
2ND ARGUMENT IS THE ORBITAL ANGULAR MOMENTUM
3RD ARGUMENT IS THE SPIN OF COMPOUND NUCLEUS
1ST DATAWORD IS THE STATISTICAL FACTOR GJ
2ND DATAWORD IS THE TOTAL WIDTH
3RD DATAWORD IS THE NEUTRON WIDTH
4TH DATAWORD IS THE RADIATION WIDTH
5TH DATAWORD IS THE FISSION WIDTH
6TH DATAWORD IS THE PROTON WIDTH
7TH DATAWORD IS THE ALPHA WIDTH
8TH DATAWORD IS THE INELASTIC WIDTH

UNKNOWN FUNCTION-VALUES ARE SET EQUAL TO ZERO
FURTHER INFORMATIONS ARE GIVEN IN KEDAK-NOTIZ NO. 3
```

```
COMMENT  8

1-6-70
PARAMETER OF THE CRANBERG-FISSION-SPECTRUM
CHI(E)=A*EXP(-B*E)*SINH(SQRT(C*E))

THE LAY-OUT OF THE DATA-RECORD IS AS FOLLOWS
1ST NAME IS THE NAME OF THE ISOTOPE
2ND NAME IS CHICR
ARGUMENT IS THE ENERGY
1ST DATAWORD IS A
2ND DATAWORD IS B
3RD DATAWORD IS C


COMMENT  9

1-6-70
POLYNOMIAL COEFFICIENTS FOR CALCULATION OF THE AVERAGE NUMBER
NUE OF THE PROMPT FISSION NEUTRONS AS FUNCTION OF THE ENERGY
NUE=NUE0+NUE1*E+NUE2*E**2+NUE3*E**3

THE LAY-OUT OF THE DATA-RECORD IS AS FOLLOWS
1ST NAME IS THE NAME OF THE ISOTOPE
2ND NAME IS PLNUE
1ST DATAWORD IS NUE0
2ND DATAWORD IS NUE1
3RD DATAWORD IS NUE2
4TH DATAWORD IS NUE3


COMMENT 10

1-6-70
ISOTOPIC ABUNDANCES

THE LAY-OUT OF THE DATA-RECORD IS AS FOLLOWS
1ST NAME IS THE NAME OF THE ELEMENT
2ND NAME IS ISOT3
ARGUMENT IS THE ATOMIC WEIGHT OF THE ISOTOPE
DATAWORD IS THE ABUNDANCE OF THE ISOTOPE


COMMENT 11

1-6-70
ANGULAR DISTRIBUTIONS OF ELASTICALLY SCATTERED NEUTRONS

THE LAY-OUT OF THE DATA-RECORD IS AS FOLLOWS
1ST NAME IS THE NAME OF THE ISOTOPE
2ND NAME IS SGNC
3RD NAME IS THE ENERGY OF THE INCIDENT NEUTRON IN THE
    LABORATORY SYSTEM
THE ARGUMENT IS THE COSINE OF THE SCATTERING ANGLE IN THE
    CENTER-OF-MASS SYSTEM
THE DATAWORD IS THE VALUE OF THE DIFFERENTIAL ELASTIC
    SCATTERING CROSS SECTION IN BARN/STERADIAN
```

```
COMMENT 12

1-8-75
COEFFICIENT F(L) IN THE LEGENDRE-POLYNOMIAL EXPANSION OF THE
DIFFERENTIAL ELASTIC SCATTERING CROSS SECTION:
SGIC(THETA)=SGIC/4.*3.14159* S((2*L+1)*F(L,E)*P(L,COS(THETA)))
                            S=SUM                    (L=0,LM)
IN THE CENTER-OF-MASS SYSTEM

THE LAYOUT OF THE DATA RECORD IS AS FOLLOWS
1ST NAME IS THE NAME OF THE ISOTOPE
2ND NAME IS LEGNC
3RD NAME IS THE ENERGY OF THE INCIDENT NEUTRON IN THE
    LABORATORY SYSTEM
4TH NAME IS THE ORDER LM
THE ARGUMENT IS L
THE DATAWORD IS F(L)


COMMENT 13

1-8-75
BIBLIOGRAPHIC INFORMATION GIVING DATA TYPES AND ENERGY REGIONS
OF RECENT EVALUATIONS

THE LAYOUT OF THE DATA RECORD IS AS FOLLOWS
1ST NAME IS THE NAME OF THE ISOTOPE
2ND NAME IS AASTATUS
DATAWORDS: ALPHAMERIC INFORMATION
(THE DATA ITEMS ARE FORMALLY DIVIDED INTO ARGUMENT AND
 FUNCTIONAL VALUE)


COMMENT 14

1-8-75
ENERGY BOUNDARIES OF THE REGION OF RESOLVED RESONANCE
PARAMETERS GIVEN IN  RES

THE LAYOUT OF THE DATA RECORD IS AS FOLLOWS
1ST NAME IS THE NAME OF THE ISOTOPE
2ND NAME IS RANGRES
1ST DATAWORD IS THE LOWER ENERGY BOUNDARY
2ND DATAWORD IS THE UPPER ENERGY BOUNDARY
3RD DATAWORD IS THE NUMBER OF RESOLVED RESONANCES IN RES
4TH DATAWORD IS A FLAG THAT INDICATES WHETHER RESOLVED
   RESONANCE PARAMETERS OR CROSS SECTION VALUES WERE TO PREFER
    FOR GROUP CONSTANT CALCULATIONS AS FOLLOWS:
2.- CROSS SECTION VALUES ARE TO PREFER
1.- RESOLVED RESONANCE PARAMETERS ARE TO PREFER
0.- NO PREFFERENCE IS RECOMMENDED
```

```
COMMENT 15

1-8-75
ENERGY DISTRIBUTION OF SECONDARY NEUTRONS
PARAMETRIC REPRESENTATION OF ENERGY SPECTRA

THE LAYOUT OF THE DATA RECORD IS AS FOLLOWS
1ST NAME IS THE NAME OF THE ISOTOPE
2ND NAME IS THE NAME OF THE DISTRIBUTION
3RD NAME IS THE ENERGY OF THE INCIDENT NEUTRON IN THE
    LABORATORY SYSTEM
THE ARGUMENT IS AN IDENTIFICATION NUMBER K OF THE MODEL USED:
K = 1 -EVAPORATION SPECTRUM
K = 2 -MAXWELLIAN SPECTRUM
K = 3 -WATT-CRANBERG SPECTRUM
K = 4 -EXCITATION OF DISCRETE LEVELS
1ST DATAWORD IS THE FRACTIONAL PROBABILITY THAT THE SPECTRUM
    OF TYPE K CAN BE USED AT THE INCIDENT ENERGY E
2ND DATAWORD IS :   FOR K=1,2 THE NUCLEAR TEMPERATURE THETA

                    FOR K=3   THE SPECTRUM PARAMETER A
                    FOR K=4   THE LEVEL EXCITATION ENERGY
3RD DATAWORD IS :   FOR K=1,2 A CONSTANT TO DEFINE THE PROPER
                              LIMIT FOR FINAL NEUTRON ENERGY
                    FOR K=3   THE SPECTRUM PARAMETER B
                    FOR K=4   =0

THE FOLLOWING TABLE SHOWS THE NAMES OF THE VARIOUS ENERGY
DISTRIBUTIONS

     SEDIC     OF NEUTRONS INELASTICALLY SCATTERED TO A
               CONTINUUM OF LEVELS
     SED2N     OF THE TWO NEUTRONS EMITTED BY THE (N,2N)-PROCESS
     SED3N     OF THE NEUTRONS EMITTED BY THE (N,3N)-PROCESS
     SEDF      OF FISSION NEUTRONS
     SEDFP     OF PROMPT FISSION NEUTRONS
     SEDFD     OF DELAYED FISSION NEUTRONS


COMMENT 16

1-8-75
ENERGY DISTRIBUTION OF THE THREE NEUTRONS EMITTED IN THE
(N,3N)-PROCESS

THE LAYOUT OF THE DATA RECORD IS AS FOLLOWS
1ST NAME IS THE NAME OF THE ISOTOPE
2ND NAME IS CHI3N
3RD NAME IS THE ENERGY OF THE INCIDENT NEUTRON IN THE
    LABORATORY SYSTEM
THE ARGUMENT IS THE NEUTRON OUTGOING ENERGY
1ST DATAWORD *
2ND DATAWORD  ** SPECTRA OF EMITTED NEUTRONS
3RD DATAWORD *
```

This page has been left internally blank.

Section 3

NDF

A retrieval program for the evaluated nuclear data library KEDAK for use in applications programs.

B. Krieg

Table of contents

Page

3. NDF - a retrieval program for the direct access KEDAK library III-143

3.1 The NDF retrieval mode III-144

3.2 The LDF retrieval mode III-145

3.3 The IDF retrieval mode III-146

3.4 Examples III-147

3.5 Source program listing III-150

3. 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.

3.1 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 FT01F001

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

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 to 1 by NDFNXT if the call has been successful. ARG 3 will contain the retrieved data. If no more data for the specified type resp. further name (set of further names) can be found, the return code is set 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

3.2 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

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 to 1 by LDFNXT if the call has been successful ARG 4 will contain the retrieved data. If no more data for the specified type resp. further name (set of further names) can be found, the return code is set to 0.

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.

3.3 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 3.2 a

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

ARG 1 see LDFLOC 3.2 b

ARG 2 see LDFLOC 3.2 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 firstfurther name
ARG 4 (4) : floating point values of the other further names
: of the set of the data required

ARG 4 see LDFLOC 3.2 b

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

ARG 1 see LDFNXT 3.2 c

ARG 2 see LDFNXT 3.2 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 3.2 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
```

3.5 Source program listing

```
C       SUBROUTINEN ZUM LESEN DER KERNDATENBIBLIOTHEK
C
C       SUBROUTINEN NDFOPN , LDFOPN ZUM EROEFFNEN DER KERNDATENBIBLIOTHEK
C
        SUBROUTINE NDF
        DIMENSION IDAT(2),IAD(1003),ISATZ(880),DAT1(60),I(4),IR(3),IW(3),
       1NNAM(4),INAM(20),KDAT1(60),MNAM(20),NUNA(2),IWNA(880),XNAM(20),
       2XJDAT(880),JDAT(880),NN(4),IBEST(2),DAT2(60),Z(60),XWNA(880)
        EQUIVALENCE (I(1),IR(1)),(Z(1),KDAT1(1)),(IWNA(1),XWNA(1)),
       1(JDAT(1),XJDAT(1))
        DATA I/'KEDA','BIBL','IOTH','    '/,NSZ/100/,NOUTP/6/
C
        ENTRY NDFOPN (LBN,IDAT,IFD,ISPR)
        JJ=1
        IDAT(1)=I(4)
        IDAT(2)=I(4)
        GOTO 50
C
        ENTRY LDFOPN (LBN,IFD,*)
        JJ=2
   50   NSZ=880
        DEFINE FILE 1 (6000,880,U,K8)
        MNAM(1)=0
        NUNA(1)=0
        IS=1
        READ (LBN'IS) (ISATZ(II),II=1,NSZ)
        IS=IS+1
        DO 1 J=1,3
        IF(ISATZ(J)-I(J))2,1,2
    2   WRITE (NOUTP,4)
    4   FORMAT(1H0/' ***ERROR NDF. 1 : THE DD-CARD FOR UNIT 1 DOES NOT CHA
       1RACTERIZE A VALID KEDAK LIBRARY')
        STOP
    1   CONTINUE
        IFD=ISATZ(4)
        IAD(1)=ISATZ(5)
        IAD(2)=ISATZ(8)
        IAD(3)=ISATZ(11)
        K=4
        IR(1)=ISATZ(6)
        IR(2)=ISATZ(9)
        IR(3)=ISATZ(12)
        IW(1)=ISATZ( 7)
        IW(2)=ISATZ(10)
        IW(3)=ISATZ( 13)
        DO 3 J=1,3
        N=IR(J)
        IWJ=IW(J)
        IF(IS-N-1)5,6,5
    5   READ (LBN'N) (ISATZ(II),II=1,NSZ)
        IS=N+1
    6   IF(J-3)10,11,11
   10   L=3
        GO TO 326
   11   L=4
```

```
  326 IMP=IAD(J)*L+IWJ-1
    8 IF(IMP-NSZ)13,14,15
   14 N=1
      GO TO 16
   15 N=2
   16 DO 12 L=IWJ,NSZ
      IAD(K)=ISATZ(L)
   12 K=K+1
      GO TO (3,17),N
   17 IMP=IMP-NSZ
      IWJ=1
      READ (LBN'IS) (ISATZ(II),II=1,NSZ)
      IS=IS+1
      GO TO 8
   13 DO318 L=IWJ,IMP
      IAD(K)=ISATZ(L)
  318 K=K+1
    3 CONTINUE
      RETURN
C
C
C     SUBROUTINEN NDFLOC , LDFLOC , IDFLOC
C
      ENTRY NDFLOC (KONTR,NNAM,DAT1,ID,KC)
      IF(NSZ.NE.880)  GO TO 1004
      ID=0
      KC=0
      DO 403 LS=1,2
      IF (DAT1(LS).NE.MNAM(LS))  GO TO 104
  403 CONTINUE
      MGL=2
      GOTO 107
  104 MGL=1
  107 J=NNAM(1)*2
      DO 51 N=1,J
   51 Z(N)=DAT1(N)
      L=1
      LS=1
      DO 18 N=1,J
      IF(N-4)19,218,20
   19 IF(N-3)20,218,218
   20 MNAM(L)=KDAT1(N)
      L=L+1
      GOTO 18
  218 IBEST(LS)=KDAT1(N)
      LS=LS+1
   18 CONTINUE
      KK=1
      GOTO427
C
      ENTRY LDFLOC (KONTR,NNAM,INAM,DAT2)
      IF(NSZ.NE.880)  GO TO 1004
      DO 127 LS=1,2
      IF (INAM(LS).NE.MNAM(LS))  GO TO 128
  127 CONTINUE
```

```
      MGL=2
      GOTO 129
  128 MGL=1
  129 J=NNAM(1)*2
      DO 21 N=1,J
   21 MNAM(N)=INAM(N)
      KK=2
  427 K=4
      IWJ=0
      DO 22 LS=1,3,2
      IWJ=IWJ+1
      N=IAD(IWJ)*3
      DO 23 M=1,N
      IF (MNAM(LS).NE.IAD(K))  GO TO 23
      IF (MNAM(LS+1).EQ.IAD(K+1))  GO TO 26
   23 K=K+3
      WRITE (NOUTP,2000) (MNAM(M),M=1,4)
 2000 FORMAT(1HO/' ***WARNING NDF. 1 : THE DATA FOR *',2A4,'*',1X,'*',
     12A4,'* ARE NOT INCLUDED'/' IN THE CONVERSIONTABLE OF THE KEDAK LIB
     2RARY')
      MNAM(1)=1
      KONTR=0
      IF(JJ-1)98,98,96
   26 NUNA(IWJ)=IAD(K+2)
   22 K=N+4
      GO TO (227,228),MGL
C
      ENTRY IDFLOC (KONTR,NNAM,INAM,DAT2)
      IF(NSZ.NE.880)  GO TO 1004
      KK=3
      IF(INAM(1)-NUNA(1))130,131,130
  131 MGL=2
      GOTO 132
  130 MGL=1
  132 NUNA(1)=INAM(1)
      NUNA(2)=INAM(2)
      IF(NNAM(1)-2)27,27,52
   52 J=NNAM(1)
      DO 53 LS=3,J
   53 MNAM(2*LS-1)=INAM(LS)
   27 GO TO (227,228),MGL
  227 N=IAD(3)*4
      IWJ=(IAD(1)+IAD(2))*3+4
      DO 28 LS=1,N,4
      IF(NUNA(1)-IAD(IWJ))28,29,28
   28 IWJ=IWJ+4
      KONTR=0
      GO TO 24
   29 NT=IAD(IWJ+1)
      JR=IAD(IWJ+2)
      JW=IAD(IWJ+3)
  228 KR=JR
      KW=JW
      IVY=0
      IF(IS-KR-1)30,31,30
```

```
30 READ (LBN'KR) (ISATZ(II),II=1,NSZ)
   IS=KR+1
31 DO 32 LS=1,NT
   IF(NUNA(2)-ISATZ(KW))33,34,33
33 KW=KW+7
   IF(KW-NSZ)32,32,35
35 READ (LBN'IS) (ISATZ(II),II=1,NSZ)
   IS=IS+1
   KW=KW-NSZ
32 CONTINUE
   KONTR=0
   GO TO 24
34 KW=KW+1
   DO 36 LS=1,6
   IF(KW-NSZ)37,37,38
38 READ (LBN'IS) (ISATZ(II),II=1,NSZ)
   IS=IS+1
   KW=1
37 GO TO (39,40,40,41,42,43),LS
39 NWN=ISATZ(KW)
   GOTO 36
40 NNAM(LS)=ISATZ(KW)
   GOTO 36
41 NWP=ISATZ(KW)
   GO TO 36
42 IDR=ISATZ(KW)
   GO TO 36
43 IDW=ISATZ(KW)
36 KW=KW+1
   IF(KK-1)46,46,47
46 NNAM(4)=0
47 DO 80 LS=1,4
80 NN(LS)=NNAM(LS)
   IF(NWN)384,384,49
49 NNK=NWP
   NWR=IDR
   NWW=IDW
   IF(IS-NWR-1)44,45,44
44 READ (LBN'NWR) (IWNA(II),II=1,NSZ)
   NWR=NWR+1
   GO TO 62
45 DO 61 LS=1,NSZ
61 IWNA(LS)=ISATZ(LS)
   NWR=NWR+1
62 DO 68 N=1,NNK
   NKO=N
   NW=NWW
   KP=5
   JD=3
   DO 54 LS=1,NWN
   IF(IWNA(NWW)-MNAM(KP))58,55,56
56 KONTR=0
   IVY=1
   IF(KK-2)57,59,859
59 INAM(KP)=IWNA(NWW)
```

```
      GOTO 60
  859 INAM(JD)=IWNA(NWW)
      GO TO 60
   57 DAT1(KP+2)=XWNA(NWW)
      GOTO 60
   55 KONTR=1
   60 KP=KP+2
      JD=JD+1
      NWW=NWW+1
      IF(NWW-NSZ)54,54,64
   64 READ (LBN'NWR) (IWNA(II),II=1,NSZ)
      NWR=NWR+1
      NWW=1
   54 CONTINUE
      GO TO 74
   58 NWW=NW+NWN+3
      IF(NWW-NSZ)68,68,70
   70 READ (LBN'NWR) (IWNA(II),II=1,NSZ)
      NWR=NWR+1
   73 NWW=NWW-NSZ
   68 CONTINUE
      KONTR=0
      IF(KK.EQ.3)  GO TO 580
      WRITE (NOUTP,2002) MNAM(5),(MNAM(II),II=1,4)
 2002 FORMAT(1H0/' ***WARNING NDF. 2 : THE FURTHER NAME ',E16.8,' IS GRE
     1ATER THAN THE GREATEST FURTHER NAME '/' INCLUDED IN THE KEDAK LIBR
     2ARY FOR',1X,2A4,1X,2A4)
      GO TO 98
  580 WRITE (NOUTP,581) MNAM(5),(INAM(II),II=1,2)
  581 FORMAT(1H0/' ***WARNING NDF. 2 : THE FURTHER NAME',E16.8,' IS GREA
     1TER THAN THE GREATEST FURTHER NAME'/' INCLUDED IN THE KEDAK LIBRAR
     2Y FOR',2I10)
      GO TO 98
 1004 WRITE (NOUTP,1005)
 1005 FORMAT(1H0/' ***ERROR NDF. 2 : AT FIRST THE --CPN - ROUTINE MUST B
     1E CALLED')
      STOP
   74 NWP=IWNA(NWW)
      DO 75 LS=1,2
      NWW=NWW+1
      IF(NWW-NSZ)76,76,78
   78 READ (LBN'NWR) (IWNA(II),II=1,NSZ)
      NWR=NWR+1
      NWW=1
   76 GO TO (81,82),LS
   81 IDR=IWNA(NWW)
      GOTO 75
   82 IDW=IWNA(NWW)
   75 CONTINUE
      NWW=NWW+1
   48 IF(NWN)384,384,383
  384 IF(IS-IDR-1)83,385,83
  383 IF(NWR-IDR-1)83,84,83
  385 DO 386 L=1,NSZ
  386 JDAT(L)=ISATZ(L)
```

```
      IDR=IDR+1
      GO TO 388
   84 DO 85 L=1,NSZ
   85 JDAT(L)=IWNA(L)
      IDR=IDR+1
      GO TO 388
   83 READ (LBN'IDR) (JDAT(II),II=1,NSZ)
      IDR=IDR+1
  388 NPA=1
   86 JD=NNAM(2)+NNAM(3)
      IF(KK-2)87,88,88
   87 LS=NNAM(1)*2+1
      GO TO 89
   88 LS=1
   89 DO 90 L=1,JD
      GO TO (91,92,92),KK
   91 DAT1(LS)=XJDAT(IDW)
      DAT1(LS+1)=0.
      LS=LS+2
      GO TO 93
   92 DAT2(LS)=XJDAT(IDW)
      LS=LS+1
   93 IDW=IDW+1
      IF(IDW-NSZ) 90,90,94
   94 READ(LBN'IDR) (JDAT(II),II=1,NSZ)
      IDR=IDR+1
      IDW=1
   90 CONTINUE
      IF(IVY)1003,1003,98
 1003 KONTR=1
      GO TO 98
   24 WRITE(NOUTP,97)NUNA(1),NUNA(2)
   97 FORMAT(1H0/' ***WARNING NDF. 3 : THE DATA FOR',2I9,' ARE NOT INCLU
     1DED IN THE KEDAK LIBRARY')
      IF(JJ-1)98,98,96
   96 RETURN 1
   98 RETURN
C
C
C     SUBROUTINEN NDFNXT, LDFNXT , IDFNXT
C
      ENTRY NDFNXT (KONTR,NNAM,DAT1,ID,KC)
      ID=0
      KC=0
      LL=1
      GO TO 101
C
      ENTRY LDFNXT(KONTR,NNAM,INAM,DAT2)
      LL=2
      GOTO 101
C
      ENTRY IDFNXT(KONTR,NNAM,INAM,DAT2)
      LL=3
  101 NPA=NPA+1
      IVY=0
      IF(NPA-NWP)102,102,103
103   KONTR=0
      IF (NWN)387,387,389
389   NKO=NKO+1
      IF(NKO-NNK)391,391,387
387   RETURN
102   KONTR=1
391   DO 304 LS=1,3
304   MNAM(LS)=NN(LS)
      IF(LL-2)105,124,106
105   MNAM(4)=0
      KDAT1(1)=MNAM(1)
      KDAT1(2)=MNAM(2)
      KDAT1(3)=IBEST(1)
      KDAT1(4)=IBEST(2)
      KDAT1(5)=MNAM(3)
      KDAT1(6)=MNAM(4)
      DO 401 II=1,6
401   DAT1(II)=Z(II)
      GO TO 125
124   DO 126 LS=1,4
126   INAM(LS)=MNAM(LS)
      GO TO 125
106   DO 327 LS=1,2
327   INAM(LS)=NUNA(LS)
125   IF(NWN)86,86,390
390   IF(KONTR)108,108,86
108   KP=NWN+3
      IVY=1
      L=5
      DO 109 LS=1,KP
      IF(NWW-NSZ)110,110,111
111   READ (LBN'NWR) (IWNA(II),II=1,NSZ)
      NWR=NWR+1
      NWW=1
110   IF(LS-NWN)112,112,113
112   IF(LL-2)114,115,116
114   L=L+2
      KDAT1(L)=IWNA(NWW)
      KDAT1(L+1)=0
      DAT1(L)=Z(L)
      DAT1(L+1)=Z(L+1)
      GO TO 109
115   INAM(L)=IWNA(NWW)
      L=L+2
      GO TO 109
116   L=LS+2
      INAM(L)=IWNA(NWW)
113   L=LS-NWN
      GO TO (119,118,117),L
119   NWP=IWNA(NWW)
      GOTO 109
118   ICR=IWNA(NWW)
      GOTO 109
117   ICW=IWNA(NWW)
109   NWW=NWW+1
      GO TO 388
      END
```

Section 4

RETPAC
A user oriented retrieval package for use with the evaluated nuclear data library KEDAK

by
R. Meyer
(Revised by E. Stein)

CONTENTS

Page

4.1. Introduction III-159

4.2. Services of RETPAC III-160

4.3. Entries III-161

4.3.1 General information III-161

4.3.2 LØCXS, NXTXS III-164

4.3.3 RETXS, REPXS III-168

4.3.4 LDAT, NDAT III-170

4.3.5 RDAT, ADAT III-173

4.3.6 LØCRES, NXTRES III-176

4.3.7 RETRES, REPRES III-179

4.3.8 RETISØ III-180

4.4. Availability III-183

4.5. Suggested Extensions III-183

4.6. References III-184

4.7. Source program listings III-186

4.1 Introduction :

Whenever data have to be stored in some computer memory, whatever type it may be, the problem will be that usually the logical structure of the data cannot be mapped directly into memory without loosing efficiency. Therefore an interface program must be available to store the physical and logical features of data and perform the respective conversions on update or retrieval. If the logical structure permitted is manifold, as it is the case with the KEDAK library, this program might be rather complicated, especially if it is desired, that the user need know as less about the actual structuring of data as possible. This retrieval package, called " library oriented retrieval package ", must be general in the sense that it must be able to handle all kind of permitted logical structure.

However , user very often deal with a specific type of data very frequently, retrieving them always in the same manner. It would therefore be desirable if a number of routines were offered that take over the task of retrieval for special types of data that are frequently used. Such a package has been constructed and is described hereafter. Similarly to the " library oriented retrieval package " it may be called user oriented retrieval package ". It may be linked to any library by the respective library oriented retrieval package, provided that the entries to be used are compatible, since it will refer only to the logical structure of the data. This aspect has been verified actually by constructing library oriented retrieval package for the transmission files of experimental data from CCDN/Saclay and for the temporary direct access storage used in CALCUL (both packages are described elsewhere), the entries of both of which are compatible with LDFPAC, IDFPAC (/1/) and NDF (/2/), the equivalent routines for KEDAK.

4.2 Services of RETPAC

The following purposes are supported by routines of RETPAC.

- Retrieval of a complete tabulated cross section type (LØCXS, NXTXS)
- Retrieval of a tabulated cross section type between a minimum and maximum energy (RETXS,REPXS).
- Both above retrieval modes with the additional feature of automatically switching to the reaction type with the logically next numerical names (if any), e.g. automatically switching from one elastic distribution to the next (LDAT,NDAT,RDAT,ADAT).
- Retrieval of data from ISØT1 - ISØT3 and automatic generation of alphameric isotope names for that material (if any) for subsequent retrieval for these isotopes (RETISØ).
- Retrieval of a complete set of resolved resonance parameters for one material (LOCRES, NXTRES).
- Retrieval of a set of resonance parameters for one material between a minimum and maximum energy (RETRES,REPRES).

All of the above routines control the filling of core storage made available and permit the user to retrieve data in sections if the amount of core storage is not sufficient to hold all data at a time.

4.3 ENTRIES

4.3.1 General information

The subsequent, detailed description of individual routines in the package are intended as a guide to KEDAK users only, since this will be the main field of application for RETPAC, and it presupposes the use of LDFPAC/1 /or NDF /2/.

This paragraph is intended as a general guide to the use of entries RETXS, REPXS, RDAT, ADAT, LOCXS, NXTXS, RETXS, REPXS and to limited extent also to LØCRES, NXTRES, RETRES, REPRES. The former entries are intended for retrieval of a tabulated, single valued function of one argument, the latter may easily be extended by a user to the more general case. It is suggested, that any routines added to the package follow the conventions outlined hereafter.

It was mentioned earlier, that the package may be linked to any library provided a suitable, that is a compatible library oriented retrieval routine package is available. This paragraph also will outline how this linkage is established.

Note that all above listed entries have one of the following argument lists.

CALL entry name (NARG,NAMES, $\underbrace{\text{(EMIN,EMAX)}}_{\text{only if applicable}}$,X, Y, NUMX, MAXNUM, NR)

- NARG and NAMES are two arrays passed directly to the library oriented retrieval routines
- EMIN and EMAX if applicable give the (energy) limits between which retrieval is to be performed.

Always, retrieval will start with the last energy $\leq$ EMIN and will stop with the first energy $\geq$ EMAX.
If interpolation to EMIN, EMAX is required, this is the user's reponsibility.

- X, Y are two arrays into which arguments and functional values are stored successively.
- NUMX is the numer of data points transmitted by the current câll.
- MAXNUM gives the maximum number of data points that may be stored into X, Y.
- NR is a returncode set by the called routine, the values of which depending upon various conditions detailed in the subsequent paragraphs.

However, the following conditions apply throughout the whole package :

NR = 1	last data point for the current data type has been stored in X, Y.
NR = 2	MAXNUM data points have been filled into X, Y without reading the end of the data type. Usually an entry is provided to continue with retrieval after a section of MAXNUM data points has been handled.
NR = 3	No data for the requested data type have been found.

The package communicates with the library by calling two entries of the form :

```
CALL        LDFLOC
CALL        LDFNXT } (NRETC,NARG,NAMES,VALUES)
```

LDFLOC is used to retrieve the first data point, LDFNXT is used to retrieve the subsequent data points.

- NRETC is a return code which is expected to be set by the called entry to 0 if no data for the requested data type have been found (LDFLOC) or if retrieval beyond the last data point of the requested data type is attempted (LDFNXT), or if any not correctable error occurs which prohibits the correct data to be transmitted. Else it is expected to be 1.

- NARG and NAMES are two arrays that are passed directly from the argument list of the respective entry of RETPAC without any changes. They are thought to contain (or receive) the logical characteristics of the data to be retrieved (NARG) and the identification and eventual further retrieval information (NAMES). They are interpreted by the library oriented retrieval routine only and need not be discussed here.

- VALUES (1) must receive the argument values and VALUES (2) the functional value of the data type to be retrieved. (For a single valued function of one argument).

Notes :

- In what follows a ↓ will denote arguments to be filled by the user, ↑ will denote arguments the values of which are returned by the routine.
- For references to material and reaction type names for KEDAK see /2/,/3/.
- All programs are written in FØRTRAN IV and have been run on IBM 36o and 37o computers.
- Storage requirements and CPU-time given apply to compilations with the H2 compiler. All programs have been tested under G, H2 and H-extended compilers.

4.3.2 LOCXS,NXTXS [x)]

Purpose : Retrieval of a complete set of data for one cross section type.

Call : CALL { LØCXS / NXTXS } (NARG,NAMES,X,Y,NUMX,MAXNUM,NR)

Arguments :

INTEGER NARG (3) : (↓) NARG (1) = number of names for reaction type requested

↑ NARG (2) = number of arguments } for that reaction type /2,3/

↑ NARG (3) = number of functional values } for that reaction type /2,3/

NARG (1) : is filled by the execution of the call if LDFPAC is used as library oriented retrieval package, else it must be filled by the user.

NARG (2) : and NARG (3) are filled by the routine and must be 1. This is not checked by the retrieval routine but is the user's responsibility.

REAL*8 NAMES (N) : ↓ NAMES (1) = material name

↓ NAMES (2) = reaction type name

↓ (↑) NAMES (3) ... NAMES (N) = eventual further, numeric names.

Replace N by the number of names necessary to uniquely describe the reaction type requested /2,3/. If applicable, numeric names must be specified by the user before executing the call. Upon

[x)] The first or only name hereafter denotes the subroutine name, all further names are names of entry points to that routine.

execution a search will be started on available numeric names for the requested reaction type. The first set of numeric names is selected, where each name is ≥ the respective numeric name given by the user. This set then is filled into NAMES (3) ... NAMES (N) and data are retrieved for this set of data.

Example : Let be NAMES (1) = 'U238', NAMES (2) = 'SGIZ', NAMES (3) =0, then the routine will return data for the first inelastic level of U238 and will insert the level energy into NAMES (3).

REAL X (MAXNUM), Y (MAXNUM) : ↑ X after execution of the call will contain the energy and

↑ Y the cross section values for the reaction type requested.

INTEGER NUM X ↑ : Number of data points read into X and Y.

INTEGER MAXNUM ↓ : Maximum number of points that may be stored into X and Y.

INTEGER NR ↑ : Return code : 1,2,3. For explanation see 4.3.1.

Notes

1) The user is responsible for opening the KEDAK library by calling LDFØPN.
2) Every time a new reaction type is to be retrieved, that is, every time one of the NAMES is changed, LOCXS must be called. If the com-

plete set of data for the requested reaction typ does not fit into arrays X,Y, a return code of 2 is set. If the program can handle the reaction type in sections, which often may be, subsequent sections of data may be retrieved by calls to NXTXS only and these calls may be repeated until a return code of 1 indicates that the last section is being transmitted.

3) Neither NARG nor NAMES may be changed between a call to LØCXS and subsequent calls to NXTXS.

Overlay : The routine is completely overlayable. If overlay is not desired, the entry NXTXS may be shortened to NXTXS (NUMX, NR), provided that neither MAXNUM nor the addresses for X, Y are changed between a call to LØCXS and subsequent calls to NXTXS.

Externals : Called routines : LDFLØC, LDFNXT - library oriented retrieval routines. See 4.3.1.

Common areas : SIGSAV - used to keep non overlayable information, must not be overlayed between a call to LOCXS and subsequent calls to NXTXS.

Storage requirements : 952 bytes.

CPU-time : depending upon the library oriented retrieval routine. For LDFPAC o.1 ms/data point on IBM 37o/165.

Example : Consider the following FORTRAN program :

```
DIMENSIØN X(1ooo),Y(1ooo),NARG(3)
REAL 8 NAMES(2)
DATA MAXNUM/1ooo/
.
.
.
```

```
                    Do not forget LDFØPN-call
                    .
                    .
                    .
                    lo   READ Names
                    .
                    .
                    .
                         GØTØ loo
                    .
                    .
                    .
                    loo  CALL LØCXS (NARG,NAMES,X,Y,NUMX,MAXNUM,NR)
                         GØTØ llo
                    lo5  CALL NXTXS (NARG,NAMES,X,Y,NUMX,MAXNUM,NR)
                    llo  IF (NR.EQ.3) GØTØ...(Error Exit : No data
                         found for that reaction type)
                         WRITE (6.6oo) (X(I), Y(I), I = 1, NUMX)
                    6oo  FØRMAT (/'DATA RETRIEVED ARE LISTED BELØW :'/
                         (1P1oE13.5) ) (Print data currently in X,Y)
                    .
                    .
                    .
                    .
                    .
                    .
                         treat problems with current section of data.
                         Number of data points is NUMX.
                    .
                    .
                    .
C----------------        ARE THERE MØRE DATA FØR CURRENT TYPE.
                         IF (NR.EQ.2) GØTØ lo5
C----------------        READ NAMES ØF NEXT REACTION TYPE.
                         GØTØ lo
                    .
                    .
                    .
                    .
                    .
```

This example shows, how a complete set of data for one reaction type can be read into main storage by sections in accordance to the length of the respective arrays, as controlled by MAXNUM. LØCXS is called first every time a new reaction type is to be retrieved. Subsequent sections for the same reaction type are retrieved by calls to NXTXS until NR no longer is 2.

4.3.3 RETXS, REPXS

Purpose : Retrieval of a set of data for one reaction type - single valued, one argument - between a lower and upper limit of argument (e.g. energy).

Call : CALL $\left\{\begin{matrix}\text{RETXS}\\\text{REPXS}\end{matrix}\right\}$ (NARG,NAMES,EMIN,EMAX,X,Y,NUMX,MAXNUM,NR)

Arguments :

REAL EMIN,EMAX ↓ : lower and upper limit to be applied. Retrieval starts with last data point with argument ≤ EMIN and stops with first data point with argument ≥ EMAX.

Integer NR ↑ : return code. 1,2,3 as explained in 4.3.1.

4... argument of first data point already > EMAX. This data point is transmitted.

5... argument of last data point still < EMIN. This data point is transmitted.

1o... transmission of data stopped because upper energy limit has been reached.

Return codes 1,2 and 1o indicate normal return, all other return codes indicate an exeptional condition.

All other arguments have been described in 4.3.2.

Notes : EMIN,EMAX must not be changed between subsequent calls to REPXS. See also 4.3.2.
Note that RETXS,REPXS are to be used similarly to LØCXS, NXTXS.

Externals : called routines : LDFLØC,LDFNXT
Common areas : SIGSAV
see also 4.3.2.

storage requirements : 1584 bytes

CPU-time : see 3.2

Example :

Consider the followings FORTRAN program

```
   DIMENSION X(1ooo),Y(1ooo),NARG(3)
   REAL x 8 NAMES (2)
   DATA MAXNUM /1ooo/
   .
   .
   Don't forget call to LDFØPN.
   .
   .
   .
   1o  Read Names, Emin, Emax.
   .
   .
   .
   .
       Goto 1oo
   1oo CALL RETXS(NARG,NAMES,EMIN,EMAX,X,Y,NUMX,MAXNUM,NR)
       GØTØ 11o
   1o5 CALL REPXS (......)
   11o IF (NR.EQ.3) GØTØ ...(Error Exit : No data can be trans-
                              mitted for that reaction type)
       IF (NR.EQ.4) GØTØ ...(Error Exit : No data in interval
                             ( EMIN,EMAX) )
```

```
IF (NR.EQ.5) GØTØ ...(Error Exit)
.
.
.
IF (X (1).GT.EMIN) GØTØ 12o
     interpolate to EMIN
.
.
12o IF (X (NUMX).LT.EMAX) GØTØ 13o
.
.
interpolate to EMAX
.
.
13o
.
.
.
treat problem
.
.
IF (NR.EQ.2) GØTØ 1o5 (read next section of data, if
   necessary)
.
.
GØTØ 1o                          (read next NAMES,EMIN,EMAX)
```

This example shows, how a set of data for one reaction type between a lower and upper energy limit may be read into main storage and sectioned according to the length of the respective arrays as indicated by MAXNUM. It also shows how error exits could be provided and how interpolation to EMIN,EMAX could be performed.

4.3.4 LDAT,NDAT

Purpose : Retrieval of a complete set of data for one cross section type with automatic switching to logical next set of numerical names. (See also example)

Call :   CALL $\left\{\begin{matrix}\text{LDAT}\\\text{NDAT}\end{matrix}\right\}$ (NARG,NAMES,X,Y,NUMX,MAXNUM,NR)

Arguments: Arguments are the same as those described in 4.3.1 except

INTEGER   NR ↑ : return code. Values 1,2,3 as described in 4.3.1, but a negative return code in addition indicates, that still another set of numerical names is available. Its data will be transmitted and its names will be filled into NAMES by a subsequent call of NDAT.(See also example).

Notes :

1) Every time a new reaction type is to be retrieved, LDAT must be used to fetch the first section of data. Subsequent sections, the availability of which is indicated by a return code of 2, can be retrieved by calls to NDAT.

2) If a reaction types requires specification of numeric names, the first set of numeric names for which data are retrieved are selected as indicated in 4.3.1. Data for subsequent sets of numeric names can be retrieved by execution of a NDAT, after a negative return code has been returned. The new numeric names will be read into NAMES and the first section of data for this set of numeric names are retrieved.

3) If numeric names apply to a reaction type, execution of a call of NDAT after a negative return code has been met,will cause a change of NAMES.

Overlay : See 4.3.2

Externals : called routine :LDFLØC, LDFNXT - See 4.3.2
Common areas : SIGSAV - See 4.3.2

LDATC - used to keep non overlayable information. The same rule as for SIGSAV applies (See 4.3.2).

Storage requirements : 1o72 bytes

CPU-time : See 4.3.2

Example :

Consider the following FORTRAN program :

```
        DIMENSION X(1o1),Y(1o1),NARG(3)
        REAL*8  NAMES(3),SGNC/'SGNC'/,MAT
        DATA MAXNUM/1o1/
        REAL LØWLIM
C------ PRØGRAM FØR PRINTØUT ØF ELASTIC DISTRIBUTIONS BETWEEN.
C------ LØWLIM AND UPLIM.
C------ A CØMPLETE DISTRIBUTION MUST BE IN CØRE AT A TIME.
        KØUT = 6
        CALL LDFØPN (1,IDAT,&999)
        NAMES (2) = SGNC
        .
        .
   1o   Read Material (MAT),LØWLIM,UPLIM
        .
        .
        NAMES (1) = MAT
        NAMES (3) = LØWLIM (Start with first distribution above Lowlim)
        .
        .
        initialize pagecounter, linecounter. Print Materialname,
        current date, date of library version (IDAT) /2/.
        .
        .
        CALL LDAT (NARG,NAMES,X,Y,NUMX,MAXNUM,NR)
        GØTØ 11o
  1oo   CALL NDAT (NARG,NAMES,X,Y,NUMX,MAXNUM,NR)
C-----  Determine error exit.
```

```
11o    IF(NR.EQ.3) GØTØ --(No elastic distribution for that
       material)
       IF (NAMES (3).GT.UPLIM) GØTØ - 2oo (Upper limit encountered).
       IF (NR.EQ.2)  GØTØ --- (Case number of data points for one
       distributation > 1oo cannot be treated).
       .
       .
       .
       .
       Print NAMES (3) (incident energy) and distributation (X(I),
            Y(I), I = 1, NUMX).
       Control page and line spacings.
       .
       .
       IF (NR.LT.0) GØTØ 1oo (fetch next distribution)
C----  No further distributions for this material
  2oo  .
       .
       .
       Print footnotes etc.
       GØTØ 1o (read name of next material)
       .
       .
       .
```

This example shows, how elastic distributions for one material in a given range of incident energies could be printed and how the return code could be used to detect overflow of arrays.

4.3.5 RDAT,ADAT

Purpose : Retrieval of a set of data for a given reaction type between a lower and upper limit of arguments.

If numeric names apply to the reaction type, automatic switching to logical next sets of numeric names are provided.

Call : CALL $\left\{ \begin{matrix} \text{RDAT} \\ \text{ADAT} \end{matrix} \right\}$ (NARG,NAMES,EMIN,EMAX,X,Y,NUMX,
MAXNUM,NR)

Arguments: See 4.3.3, except that a negative return code indicates, that another set of numeric names is available for that reaction type. Upon a subsequent call to ADAT the first section for this set of numeric names of the current reaction type will be retrieved and the new numeric names are filled into NAMES as outlined already in the preceding paragraph.

Notes : See 4.3.4

Overlay : See 4.3.2

Externals : Called routines : LDFLØC,LDFNXT - see 4.3.1
common areas : SIGSAV and LDATC - see 4.3.4

storage requirements : 1772 bytes

CPU-time : See 4.3.2

Example :

The following program shows, how data for all inelastic levels of a material within a given energy range could be retrieved (again sectioning of data is provided).

```
      DIMENSIØN X (5oo), Y (5oo), NARG (3)
      REAL*8   NAMES (3)
      DATA MAXNUM /5oo/, NAMES (2) /'SGIZ'/
      .
      .
      .
      .
      Don't forget call to LDFØPN
      .
      .
  1o  Read Names (1), Emin,Emax
      Names (3) = 0 (Start with first level).
      .
      .
      .
      .
      CALL RDAT (NARG,NAMES,EMIN,EMAX,X,Y,NUMX,MAXNUM,
      NR) GØTØ 11o
 1oo  CALL ADAT (.....)
 11o  IF (NR.EQ.3) GØTØ... (No inelastic level excitation
      data for that material).
      .
      .
      .
      .
      WRITE (6.601) NAMES(3), (X(I), Y(I), I = 1,NUMX)
 6o1  FORMAT (/'CURRENT DATA FØR LEVEL', 1PE13.5/(1oE13.5) )
      .
      .
      .
      treat problem
      .
      .
      .
      .
      IF (NR.EQ.2) GØTØ 1oo (fetch next section)
      IF (NR.LT.O) GØTØ 2oo (this level finished)
      GØTØ 1o (read next material name, etc..)
  2oo .
      .
      .
      .
```

```
perform eventual closing calculations
.
.
GØTØ loo (Fetch first section of data for next level)
.
.
.
```

4.3.6 LØCRES,NXTRES

Purpose : Retrieve a complete set of resonance for one material.

Call : CALL $\left\{\begin{matrix}\text{LØCRES}\\ \text{NXTRES}\end{matrix}\right\}$ (NR,MAT,MAXNUM,X,NUMX)

Arguments :

INTEGER NR ↑ : Return code 1,2, or 3 as described in 4.3.1

REAL*8 MAT ↓ : Name of material, for which resonances shall be retrieved.

INTEGER MAXNUM↓ : Maximum number of resonance that can be read into X at a time.

REAL X (MAXNUM, 11) ↑ : Upon successful completion of the call X will contain the resonance parameters, as given in references /2,3/.

INTEGER NUMX ↑ : Number of resonances read into X by the current execution of the call.

Notes :

1) the dimensions of X have been chosen so that by suitable equivalencing a set of a given resonance quantity (e.g.resonance energy of Γ_{tot}) can be referred to by a single dimensioned array (see example). The second dimension of 11 has been chosen in accordance to the number of resonance parameters given for each resonance (see ref. /2,3/)

2) Every time resonance parameters for a new material are requested, the first section of data is retrieved by a call to LØCRES, while subsequent sections may be retrieved by NXTRES if a returncode of 2 is encountered.

3) See note 3 of 4.3.2

Overlay : The routine is fully overlayable. If overlay is not desired, the entry NXTRES may be shortened to NXTRES (NR,NUMX), provided that MAXNUM and the other addresses are not changed between subsequent executions of NXTRES.

Externals : Called routines : LDFLØC, LDFNXT - See 4.3.1
Common areas : RESAVE - used to keep non overlayable information. Must not be overlaid between retrievals of subsequent sections of resonance parameters for the same material.

storage requirements : 812 bytes

CPU-time : about o.3 ms/resonance on IBM 37o/165 if used with LDFPAC/1/.

Example : Consider the following FORTRAN program :

```
 DIMENSION X (1oo,11),ERES(1),GJ(1),GAMTØT(1),
 GAMN(1),GAMG(1),GAMF(1)
 REAL*8 MAT
 DATA MAXNUM/1oo /
 EQUIVALENCE (X(1,1),ERES(1),(X(1,4),GJ(1) ),(X(1,5),
*GAMTØT(1) ), (X(1,6),GAMN(1) ), (X(1,7),GAMG(1) ),(X(1,8),
 GAMF (1) )
 .
 .
 .
 .
 Don't forget LDFØPN-Call
```

```
        .
        .
        .
1o      Read MAT (material name)
        .
        .
        .
        CALL LØCRES (NR,MAT,MAXNUM,X,NUMX)
        GØTØ 11o
1oo     CALL NXTRES (NR,MAT,MAXNUM,X,NUMX)
11o     IF(NR.EQ.3) GØTØ ... (Error Exit.No resonance parameters
        for this material)
        WRITE (6,6oo)MAT,(ERES(I),GJ(I),GAMTØT(I),GAMN(I),
        GAMG(I),GAMF(I), I=1, NUMX)
6oo     FØRMAT ('RESONANCE DATA FØR MATERIAL',A8//'RES.ENERGY',
               T14,'GJ',T27,'GAM.TØTAL',T4o,'GAM.ELASTIC',
               T53,'GAM.GAMMA',T66,'GAM.FISS.'/(1P6E13.5.) )
        .
        .
        .
```

treat problem.

You may refer to resonance energy by ERES, to statistical weight factor by GJ, to Γ_{tot} by GAMTØT, to Γ_{n} by GAMN, to Γ_{γ} by GAMG and to Γ_{f} by GAMF.

```
        .
        .
        .
        IF (NR.EQ.2) GØTØ 1oo (fetch next section of resonance, if number
        of resonance > 1oo)
        GØTØ 1o (read next material name).
        .
        .
        .
        .
```

4.3.7 RETRES, REPRES

Purpose : Retrieve a set of resolved resonance parameters for a given material between given energy limits.

Call : CALL $\left\{ \begin{matrix} \text{RETRES} \\ \text{REPRES} \end{matrix} \right\}$ (NR,MAT,MAXNUM,X,NUMX,EMIN,EMAX)

Arguments :

real EMIN,EMAX ↓ : lower and upper energy limits. Retrieval starts with last resonance $\leq$ EMIN and stops with first resonance $\geq$ EMAX.

Integer NR ↑ 1,2,3,4,5,1o as described in 4.3.3
For the other arguments see 4.3.6

Notes : See 4.3.6

Overlay : See 4.3.6

Externals : Called routines : LDFLØC,LDFNXT - see 4.3.1
Common areas : RESAVE - to keep non overlayable information. Must not be overlaid between retrieval of subsequent sections of resonance parameters for the same material.

storage requirements : 1872 bytes

CPU-time : o.3 ms/resonance on IBM 37o/165 if used with LDFPAC/1/

Example : See 4.3.6. Example given there can be modified directly to take into account EMIN,EMAX.See also 4.3.3.

4.3.8 RETISØ

Purpose : Retrieve information in ISØT1 - ISØT3 and prepare alphamerical material names of isotopes, if any available for given element.

Call : CALL RETISØ (MAT,A,Z,I,LAMBDA,RADIUS,EBIND,AISØ,ABUN,MATISØ, NISØ,MAXISØ,IERR)

Arguments :

REAL*8 ↓ : name of material, for which information shall be retrieved.

REAL A,Z,I ↑ : Atomic weight, charge number, ground state spin (all from ISØT1/2/).

REAL LAMBDA, RADIUS,EBIND ↑ : λ , nuclear radius, binding energy of last neutron (all from ISØT2/2/).

REAL AISØ (MAXISØ),ABUN (MAXISØ) ↑ : if material is an element and isotopic information is available, AISØ gives the atomic mass numbers of the isotopes and ABUN the respective isotopic natural abundancy (from ISØT3 /2/).

REAL*8 (MAXISØ) ↑ : Alphameric names of available isotopes (if any). Is constructed from AISØ and MAT according usual KEDAK-conventions, and may be used for further retrieval of data for isotopes.

INTEGER NISØ ↑ : number of isotopes, for which data have been stored into AISØ, ABUN, MATISØ (if any), 0 else. Note that NISØ ≤ MAXISØ always.

INTEGER MAXISØ ↓ : maximum number of isotopes, for which space is reserved in AISØ, ABUN and MATISØ.

INTEGER IERR (3)

↑ : IERR (1) 0 else
1 no ISØT1 available

IERR (2) — 0 else
1 no ISØT2 available

IERR (3) = number of isotopes found (even if > MAXISØ).
IERR (3) > NISØ indicates insufficient dimensions for arrays AISØ, ABUN,MATISØ.

Overlay : Routine RETISØ is fully overlayable

Externals : Called routines : LDFLØC,LDFNXT - see4.3.1
Convy - to convert atomic mass numbers to character string representation to generate array MATISØ. /4/
common areas : none

storage required : 1146 bytes

CPU-time : negligible

Example : the following FORTRAN program shows, how ISØT-information could be retrieved for an element and all its isotopes :

```
DIMENSION AISØ ( 1o ), ABUN (1o), MATISØ (1o), IERR (3), DUMMY (1o)
REAL*8 MATISØ,MAT,DUMMY
REAL LAMBDA, I
DATA MAXISØ /1o/
   .
   .
Do not forget LDFØPN-Call
   .
   .
   .
   .
   .
1o Read materialname MAT
   .
   .
C ----  Retrieve DATA FØR MAT.
  CALL RETISØ (MAT,A,Z,I,LAMBDA,RADIUS,EBIND,AISØ,ABUN,MATISØ,
  NISØ,MAXISØ,IERR)
  Test IERR (1), IERR (2)
  treat problem
C ---- RETRIEVE DATA FOR ISOTØPES,IF ANY.
IF (NIS.EQ.O) GOTO 1o (no isotopes)
IF (IERR (3).GT.NIS)....(Print warning message : Dimensions of
AISØ,ABUN,MATISØ too small.
DØ 1oo I = 1, NISØ
CALL RETISØ (MATISØ (1), A,Z,I,LAMBDA,RADIUS,EBIND,DUMMY,DUMMY,
DUMMY,DUMMY,MAXISØ,IERR)
(Dummies are used since for an isotope no ISØT3 should be available.
For security reason, however, DUMMY has been dimensioned).
   .
   .
   .
   .
test IERR (1), IERR (2)
   .
   .
   .
   .
treat problem
   .
   .
   .
1oo CØNTINUE
   .
   .
   .
   .
```

4.4 AVAILABILITY

RETPAC is available in two forms:

- The complete source can be made available by the nuclear data evaluation group of the INR.

- The package is available also as load module from INR.STEIN.LOAD GFK029 as member RETPAC. Individual routines may be isolated by using the REPLACE statement of the linkage editor.

4.5 SUGGESTED EXTENSIONS

A simple exstension of LØCXS,NXTXS or RETXS,REPXS could be programmed by extending the routines to a variable number of arguments and functional values. It is suggested, that a single array is used to transmit the data points, which is dimensioned similar to array X of routine LØCRES. That is, the number of arguments and functional values should be the second dimension to facilitate simple addressing of individual data items by equivalencing as is demonstrated in the example of 4.3.6. Also EMIN and EMAX will be arrays in this case.

Another extension has already been verified, when routines of RETPAC were entered into the calculation routine package /5/. LDFLØC and LDFNXT were entered via argument list, also and three different types of library thus could be linked to the same user oriented retrieval package : KEDAK, an intermediate direct access library for automatic roll out of data that cannot be held in main storage, and the transmission files for experimental data distributed by Saclay /6/.

4.6 REFERENCES

/1/ I. Langner, R. Meyer
IDFPAC - LDFPAC
Institut für Neutronenphysik und Reaktortechnik,
Gesellschaft für Kernforschung mbH., Karlsruhe
KFK 2387/III, Section 2

/2/ B. Krieg
Handling and Service programs for the Karlsruhe Nuclear
Data File KEDAK
Part I: Management and Retrieval Programs
Institut für Neutronenphysik und Reaktortechnik,
Gesellschaft für Kernforschung mbH., Karlsruhe
KFK 1725 Juni 1973

/3/ D. Woll
Card Image Format of the Karlsruhe Evaluated Nuclear
Data File KEDAK.
Institut für Neutronenphysik und Reaktortechnik
Gesellschaft für Kernforschung mbH., Karlsruhe
KFK 880 (EANDC(E)-112"U"EUR 4160e)
Dezember 1968

/4/ H. Blesene
CONVY, FORTRAN-Unterprogramm für die IBM/360 zur
Umwandlung von in maschineninterner bzw. in alphanumerischer
Darstellung vorliegenden Fest- und Gleitkommazahlen
in alphanumerische bzw. maschineninterne Darstellung.
Not published.

/5/ I. Langner, R. Meyer
CALCUL.
A program to calculate composed cross sections for the
KEDAK library.
Institut für Neutronenphysik und Reaktortechnik,
Gesellschaft für Kernforschung mbH., Karlsruhe
KFK 2387/VII

/6/ Neudada System Description
Report of the Nuclear Energy Agency
CCDN/SYS-2, April 1969

4.7 Source program listings

```
      SUBROUTINE LOCXS(NARG,NAMES,X,Y,NUMX,MAXNUM,NR)
      COMMON/SIGSAV/ Z
      DIMENSION X(1),Y(1),Z(2),NARG(1)
      REAL*8 NAMES(1),NAMSV(4)
C
C         LOCXS : ARE RETRIEVAL SUBROUTINES
C         NXTXS : FOR KEDAK DATA.
C
      I=0
      CALL LDFLOC(NERR,NARG,NAMES,Z)
      IF(NERR.EQ.0) GOTO 30
   19 NAMZ=NARG(1)
      IF(NAMZ.LE.2) GOTO 21
      DO 10 J=3,NAMZ
   10 NAMSV(J-2)=NAMES(J)
      GOTO 21
   20 CALL LDFNXT(NERR,NARG,NAMES,Z)
      IF(NERR.EQ.0) GOTO 22
      IF(I.EQ.MAXNUM) GOTO 34
   21 I=I+1
      X(I)=Z(1)
      Y(I)=Z(2)
      GOTO 20
C
      ENTRY NXTXS(NARG,NAMES,X,Y,NUMX,MAXNUM,NR)
      I=0
      GOTO 19
C
   22 NR=1
      IF(NAMZ.LE.2) GOTO 200
      DO 23 J=3,NAMZ
   23 NAMES(J)=NAMSV(J-2)
      GOTO 200
   30 NR=3
      GOTO 200
   34 NR=2
      GOTO 200
  200 NUMX=I
      RETURN
      END
      SUBROUTINE RETXS(NARG,NAMES,EMIN,EMAX,X,Y,NUMX,MAXNUM,NR)
      COMMON/SIGSAV/Z
      DIMENSION X(1),Y(1),Z(2),NARG(1),W(2)
      REAL*8 NAMES(1),NAMSV(4)
C
C          RETXS RETRIEVES KEDAK-DATA.
C
      ASSIGN 20 TO NST
      I=0
      CALL LDFLOC(NERR,NARG,NAMES,Z)
      IF(NERR.EQ.0) GOTO 30
      NAMZ=NARG(1)
      IF(NAMZ.LE.2) GOTO 3
      DO 2 J=3,NAMZ
    2 NAMSV(J-2)=NAMES(J)
   3  IF(Z(1).LE.EMIN) GOTO 5
      IF(Z(1).GE.EMAX) GOTO 32
      GOTO 21
    5 CALL LDFNXT(NERR,NARG,NAMES,W)
      IF(NERR.EQ.0) GOTO 23
      IF(W(1).LE.EMIN) GOTO 10
      IF(W(1).GE.EMAX) GOTO 36
   7  I=I+1
      X(I)=Z(1)
      Y(I)=Z(2)
      I=I+1
      X(I)=W(1)
      Y(I)=W(2)
      GOTO NST,(20,200)
   10 CALL LDFNXT(NERR,NARG,NAMES,Z)
      IF(NERR.EQ.0) GOTO 26
      IF(Z(1).LE.EMIN) GOTO 5
      IF(Z(1).GE.EMAX) GOTO38
   11 I=I+1
      X(I)=W(1)
      Y(I)=W(2)
   12 I=I+1
      X(I)=Z(1)
      Y(I)=Z(2)
      GOTO NST,(20,200)
   20 CALL LDFNXT(NERR,NARG,NAMES,Z)
      IF(NERR.EQ.0) GOTO 22
      IF(Z(1).GE.EMAX) GOTO 24
      IF(I.EQ.MAXNUM) GOTO 34
   21 I=I+1
      X(I)=Z(1)
      Y(I)=Z(2)
      GOTO 20
C
      ENTRY REPXS(NARG,NAMES,EMIN,EMAX,X,Y,NUMX,MAXNUM,NR)
      I=0
      NAMZ=NARG(1)
      IF(NAMZ.LE.2) GOTO 21
      DO 19 J=3,NAMZ
   19 NAMSV(J-2)=NAMES(J)
      GOTO 21
C
   22 IF(I.LT.1) GOTO 23
      NR=1
      GOTO 198
   23 NR=5
      I=1
      X(I)=Z(1)
      Y(I)=Z(2)
      GOTO 198
C
   24 IF(I.EQ.MAXNUM) GOTO 34
      I=I+1
      X(I)=Z(1)
      Y(I)=Z(2)
      NR=10
      GOTO 200
   26 I=1
      X(I)=W(1)
      Y(I)=W(2)
      NR=5
      GOTO 198
C
   30 NR=3
      GOTO 200
C
   32 NR=4
      I=1
      X(1)=Z(1)
      Y(1)=Z(2)
      GOTO 200
C
   34 NR=2
      GOTO 200
   36 ASSIGN 200 TO NST
      NR=10
      GOTO 7
   38 ASSIGN 200 TO NST
      NR=10
      GOTO 11
  198 IF(NAMZ.LE.2) GOTO 200
      DO 199 J=3,NAMZ
  199 NAMES(J)=NAMSV(J-2)
  200 NUMX=I
      RETURN
      END
      SUBROUTINE LDAT(NARG,NAMES,X,Y,NUMX,MAXNUM,NR)
      COMMON/SIGSAV/ Z
      COMMON/LDATC/NAMSV,NAMNXT
      REAL*8 NAMES,NAMNXT,NAMSV
      DIMENSION X(1),Y(1),NARG(1),NAMES(1),NAMNXT(6),NAMSV(6),Z(2)
C
C         LDAT,NDAT FOR RETRIEVAL OF SINGLE VALUED KEDAK DATA WITH
C         MORE THAN TWO NAMES.
      GOTO 50
   10 IF(NAMZ.LE.2) GOTO 100
      DO 20 I=1,NAMZ
      IF(NAMNXT(I).EQ.NAMES(I)) GOTO 20
      GOTO 30
   20 CONTINUE
      GOTO 100
   30 NR=-NR
      GOTO 100
C
      ENTRY NDAT(NARG,NAMES,X,Y,NUMX,MAXNUM,NR)
      NAMZ=NARG(1)
      IF(NR.GE.0) GOTO 72
      DO 40 I=1,NAMZ
   40 NAMES(I)=NAMNXT(I)
      GOTO 50
C
   50 J=0
      CALL LDFLOC(NERR,NARG,NAMES,Z)
      IF(NERR.EQ.0) GOTO 80
      NAMZ=NARG(1)
      DO 55 I=1,NAMZ
   55 NAMSV(I)=NAMES(I)
      GOTO 71
   70 CALL LDFNXT(NERR,NARG,NAMES,Z)
      IF(NERR.EQ.0) GOTO 75
   71 IF(J.EQ.MAXNUM) GOTO 85
      J=J+1
      X(J)=Z(1)
      Y(J)=Z(2)
      GOTO 70
   72 J=0
      GOTO 71
C
   75 NR=1
      DO 76 I=1,NAMZ
      NAMNXT(I)=NAMES(I)
   76 NAMES(I)=NAMSV(I)
      GOTO 10
   80 NR=3
      GOTO 100
   85 NR=2
      GOTO 100
  100 NUMX=J
      RETURN
      END
      SUBROUTINE RETISO(MAT,A,Z,I,LAMBDA,R,EB,AISO,ABUN,MATISO,NIS,MAXIS
     1,IERR)
      REAL*8 MAT,NAMES(2),ISOT1/'ISOT1'/,ISOT2/'ISOT2'/,ISOT3/'ISOT3'/,
     1 NAM,MATISO(1)
      DIMENSION NARG(3),FW(3),AISO(1),ABUN(1),IERR(1)
      DATA NARG/2,2*0/
      REAL I,LAMBDA
      LOGICAL*1 LNAM(8)
      EQUIVALENCE (LNAM,NAM)
C
C         RETISO RETRIEVES ISOT1-ISOT3 DATA.
C
C         ISOT1.
      NAMES(1)=MAT
      NAMES(2)=ISOT1
      CALL LDFLOC(NR,NARG,NAMES,FW)
      IF(NR.NE.0) GOTO 10
      IERR(1)=0
      GOTO 20
   10 A=FW(1)
      Z=FW(2)
      I=FW(3)
      IERR(1)=1
C
C         ISOT2.
   20 NAMES(2)=ISOT2
      CALL LDFLOC(NR,NARG,NAMES,FW)
      IF(NR.NE.0) GOTO 30
      IERR(2)=0
      GOTO 40
   30 LAMBDA=FW(1)
      R=FW(2)
      EB=FW(3)
      IERR(2)=1
C
C         ISOT3
   40 NAMES(2)=ISOT3
      NIS=0
      IERR(3)=0
      CALL LDFLOC(NR,NARG,NAMES,FW)
      IF(NR.NE.0) GOTO 50
      GOTO 100
   50 NIS=NIS+1
      IERR(3)=IERR(3)+1
      AISO(NIS)=FW(1)
      ABUN(NIS)=FW(2)*.01
      ISAG=FW(1)
      XISAG=ISAG
      IF(FW(1)-XISAG.GE.0.5) ISAG=ISAG+1
      NAM=MAT
      CALL CONVY(ISAG,LNAM(3),3HI3 ,1H0)
      MATISO(NIS)=NAM
      CALL LDFNXT(NR,NARG,NAMES,FW)
      IF(NR.EQ.0) GOTO 100
      IF(NIS.NE.MAXIS) GOTO 50
      IERR(3)=IERR(3)+1
   60 CALL LDFNXT(NR,NARG,NAMES,FW)
      IF(NR.EQ.0) GOTO 100
      IERR(3)=IERR(3)+1
      GOTO 60                                                           00002550
  100 RETURN                                                            00002560
      END                                                               00002570
      SUBROUTINE LOCRES(NERR,MAT,LENX,X,MAX)
      COMMON/RESAVE/Z,NARG
      DIMENSION X(LENX,1),NARG(3),NAMEN(2),Z(20)
      REAL*8 MAT,NAMEN
      DATA NAMEN(2)/'RES'/
C         LOCRES AND NXTRES RETRIEVE RESONANCE DATA WITHOUT CONTROL
C         ON MINIMUM OR MAXIMUM ENERGY.
C         USE LOCRES FOR FIRST ENTRY,NXTRES FOR THE FOLLOWING ONES.
C
      NARG(1)=2
      I=0
      NAMEN(1)=MAT
      I=0
      CALL LDELOC(NR,NARG,NAMEN,Z)
      IF(NR.EQ.0) GOTO 30
      LLX=NARG(2)+NARG(3)
      GOTO 25
   20 CALL LDENXT(NR,NARG,NAMEN,Z)
      IF(NR.EQ.0) GOTO 34
      IF(I.EQ.LENX) GOTO 36
   25 I=I+1
      DO 26 K=1,LLX
   26 X(I,K)=Z(K)
      GOTO 20
C
      ENTRY NXTRES(NERR,MAT,LENX,X,MAX)
      LLX=NARG(2)+NARG(3)
      I=0
      GOTO 25
C
   30 NERR=3
      GOTO 100
C
   34 NERR=1
      GOTO 100
C
   36 NERR=2
      GOTO 100
C
  100 MAX=I
      RETURN
      END
      SUBROUTINE RETRES(NERR,MAT,LENX,X,MAX,EMIN,EMAX)
      COMMON/RESAVE/Z,NARG
      DIMENSION X(LENX,1),NARG(3),NAMEN(2),Z(20),W(20)
      REAL*8 MAT,NAMEN
      DATA NAMEN(2)/'RES'/
C
C RETRES RETRIEVES RESONANCE PARAMETERS FROM KEDAK FOR RESONANCE
C         ENERGIES BETWEEN EMIN,EMAX AND PUTS THEM INTO X.
C         A MAXIMUM NUMBER LENX OF RESONANCES IS ACCEPTED.
C
      NARG(1)=2
      I=0
      ASSIGN 20 TO NST
      NAMEN(1)=MAT
      CALL LDFLOC(NR,NARG,NAMEN,Z)
      IF(NR.EQ.0) GOTO 30
      LLX=NARG(2)+NARG(3)
      IF(Z(1).LE.EMIN) GOTO 5
      IF(Z(1).GE.EMAX) GOTO 32
      GOTO 25
    5 CALL LDFNXT(NR,NARG,NAMEN,W)
      IF(NR.EQ.0) GOTO 34
      IF(W(1).LE.EMIN) GOTO 10
      IF(W(1).GE.EMAX) GOTO 42
   8  I=I+1
      DO 6 K=1,LLX
    6 X(I,K)=Z(K)
      I=I+1
      DO 7 K=1,LLX
    7 X(I,K)=W(K)
      GOTO NST,(20,100)
   10 CALL LDFNXT(NR,NARG,NAMEN,Z)
      IF(NR.EQ.0) GOTO 37
      IF(Z(1).LE.EMIN) GOTO 5
      IF(Z(1).GE.EMAX) GOTO 44
   13 I=I+1
      DO 11 K=1,LLX
   11 X(I,K)=W(K)
      I=I+1
      DO 12 K=1,LLX
   12 X(I,K)=Z(K)
      GOTO NST,(20,100)
   20 CALL LDFNXT(NR,NARG,NAMEN,Z)
      IF(NR.EQ.0) GOTO 35
      IF(Z(1).GE.EMAX) GOTO 39
      IF(I.EQ.LENX) GOTO 36
   25 I=I+1
      DO 26 K=1,LLX
   26 X(I,K)=Z(K)
      GOTO 20
C
      ENTRY REPRES(NERR,MAT,LENX,X,MAX,EMIN,EMAX)
      LLX=NARG(2)+NARG(3)
      I=0
      GOTO 25
C
   30 NERR=3
      GOTO 100
C
```

```
   32 NERR=4
   31 I=1
      DO 33 K=1,LLX
   33 X(I,K)=Z(K)
      GOTO 100
C
   34 NERR=5
      GOTO 31
   35 NERR=1
      GOTO 100
C
   36 NERR=2
      GOTO 100
   37 NERR=5
      I=1
      DO 38 K=1,LLX
   38 X(I,K)=W(K)
      GOTO 100
C
   39 IF(I.EQ.MAXNUM) GOTO 36
      I=I+1
      DO 40 K=1,LLX
   40 X(I,K)=Z(K)
      NERR=10
      GOTO 100
   42 ASSIGN 100 TO NST
      NERR=10
      GOTO 8
   44 ASSIGN 100 TO NST
      NERR=10
      GOTO 13
  100 MAX=I
      RETURN
      END
      SUBROUTINE RDAT(NARG,NAMES,EMIN,EMAX,X,Y,NUMX,MAXNUM,NR)
      COMMON/SIGSAV/ Z
      DIMENSION X(1),Y(1),Z(2),NARG(1),W(2)
      COMMON/LDATC/NAMSV,NAMNXT
      REAL*8 NAMES(1),NAMNXT(6),NAMSV(6)
C
C         RDAT,ADAT RETRIEVE KEDAK-DATA WITH MORE THAN TWO NAMES.
C                (SINGLE VALUED).
      GOTO 50
   10 IF(NAMZ.LE.2) GOTO 100
      DO 20 I=1,NAMZ
      IF(NAMNXT(I).EQ.NAMES(I)) GOTO 20
      GOTO 30
   20 CONTINUE
      GOTO 100
   30 NR=-NR
      GOTO 100
C
      ENTRY ADAT(NARG,NAMES,EMIN,EMAX,X,Y,NUMX,MAXNUM,NR)
      NAMZ=NARG(1)
      IF(NR.GE.0) GOTO 72
      DO 40 I=1,NAMZ
   40 NAMES(I)=NAMNXT(I)
      GOTO 50
C
   50 J=0
      ASSIGN 70 TO NST
      CALL LDFLOC(NR,NARG,NAMES,Z)
      IF(NR.EQ.0) GOTO 80
      NAMZ=NARG(1)
      DO 55 I=1,NAMZ
   55 NAMSV(I)=NAMES(I)
      IF(Z(1).LE.EMIN) GOTO 60
      IF(Z(1).GE.EMAX) GOTO 82
      GOTO 67
   60 CALL LDFNXT(NR,NARG,NAMES,W)
      IF(NR.EQ.0) GOTO 73
      IF(W(1).LE.EMIN) GOTO 65
      IF(W(1).GE.EMAX) GOTO 85
   62 J=J+1
      X(J)=Z(1)
      Y(J)=Z(2)
      J=J+1
      X(J)=W(1)
      Y(J)=W(2)
      GOTO NST,(70,77)
   65 CALL LDFNXT(NR,NARG,NAMES,Z)
      IF(NR.EQ.0) GOTO 79
      IF(Z(1).LE.EMIN) GOTO 60
      IF(Z(1).GE.EMAX) GOTO 86
   66 J=J+1
      X(J)=W(1)
      Y(J)=W(2)
   67 J=J+1
      X(J)=Z(1)
      Y(J)=Z(2)
      GOTO NST,(70,77)
   70 CALL LDFNXT(NR,NARG,NAMES,Z)
      IF(NR.EQ.0) GOTO 73
      IF(Z(1).GE.EMAX) GOTO 75
   71 IF(J.EQ.MAXNUM) GOTO 84
      J=J+1
      X(J)=Z(1)
      Y(J)=Z(2)
      GOTO 70
C
   72 J=0
      GOTO 71
C
   73 NR=1
   74 DO 81 I=1,NAMZ
      NAMNXT(I)=NAMES(I)
   81 NAMES(I)=NAMSV(I)
      GOTO 10
   75 IF(J.LT.1) GOTO 76
      IF(J.EQ.MAXNUM) GOTO 84
      J=J+1
      X(J)=Z(1)
      Y(J)=Z(2)
      NR=10
      GOTO 77
   76 NR=6
      GOTO 77
   77 CALL LDFNXT(NER,NARG,NAMES,Z)
      IF(NER.EQ.0) GOTO 74
      GOTO 77
   78 J=1
      X(J)=Z(1)
      Y(J)=Z(2)
      NERR=5
      GOTO 74
   79 J=1
      X(J)=W(1)
      Y(J)=W(2)
      NR=5
      GOTO 74
   80 NR=3
      GOTO 100
   82 NR=4
      J=1
      X(J)=Z(1)
      Y(J)=Z(2)
      GOTO 77
   84 NR=2
      GOTO 100
   85  ASSIGN 77 TO NST
      NR=10
      GOTO 62
   86 ASSIGN 77 TO NST
      NR=10
      GOTO 66
  100 NUMX=J
      RETURN
      END
```