WORLD HEALTH ORGANIZATION
ORGANISATION MONDIALE DE LA SANTÉ

HSM/79.1
ORIGINAL: FRENCH

TWH (Tripartition for World Health)

A PROGRAM SYSTEM FOR THE AUTOMATIC PRODUCTION OF STATISTICAL TABLES WITH DATA FROM CROSS-SECTIONAL AND LONGITUDINAL STUDIES[1]

by

P. C. Kaufmann
World Health Organization, Geneva

Write-up in collaboration with J. Vaugelade
Organisation de la Recherche scientifique
et technique d'Outre-Mer, Paris

CONTENTS

[1] This document was produced in 1975 under reference HSM/75.2.

1. GENERAL DESCRIPTION

TWH is a program system for the processing of coded information from cross sectional or longitudinal surveys. This system has been used for the verification, correction, updating, recoding, file production, tabulation, and elementary statistical analysis of survey data, for which both input and output volumes of information were large. It has also been used in connexion with the designing of record systems. A more detailed description of the program functions is given in Chapter 3. The system included three main programs and optional programs, which may be used for particular applications. The first program RECODE reads data from cards or tape, verifies code ranges and produces recoded data. This output from RECODE (file ZZ) is sorted and organized into a hierarchical file (Z) by the GFILE program, which can also perform calculations on variables. Then the tabulation program GTAB edits frequency distribution tables and elementary statistics (see in figure 1 the sequence of operations). Two sorting operations are usually necessary. They are performed by the SORT program from IBM.

The Z file is the centre of the conception of these programs; it is a hierarchical file with three levels (for instance: household, individual, event). In figure 1 the 2nd sort is a function of this hierarchy: it should group together the individuals of the same household and the events of these individuals, so that GFILE can identify the structure of the household, define a unit of information with minimal length, calculate new variables and check the consistency of the information inside the household. For certain applications it might be necessary to attribute an element of information to different hierarchies (for instance a child might be attributed to hierarchy 2 or 3 according to the file type). This would imply various sorting operations and several processings with GFILE. Sequential utilizations of GFILE are possible, an additional unstructured ZZ file, as output of GFILE, being used as input to a subsequent GFILE. The information calculated by a first GFILE can be used by a second GFILE (see figure 2). The optional program DELRET (figure 2) deletes these records which are not relevant in the file ZZ2.

The file ZZ can, at the limit, contain only one category of information. It can be treated, if desired, as an unstructured file, the processing of which is more economical. In this case GFILE is an optional program which can be used for the calculation of new variables and/or verifications of consistency.

The other programs are optional. Three of them edit the parameter cards which are used for the three main programs, thus giving very useful information to the persons which are responsible for the processing and the analysis of the information. Other programs refer to the files. For instance REVISE is used for the correction on ZZ files, while DELRET selects certain types of records, forming a shorter file which is less expensive to process.

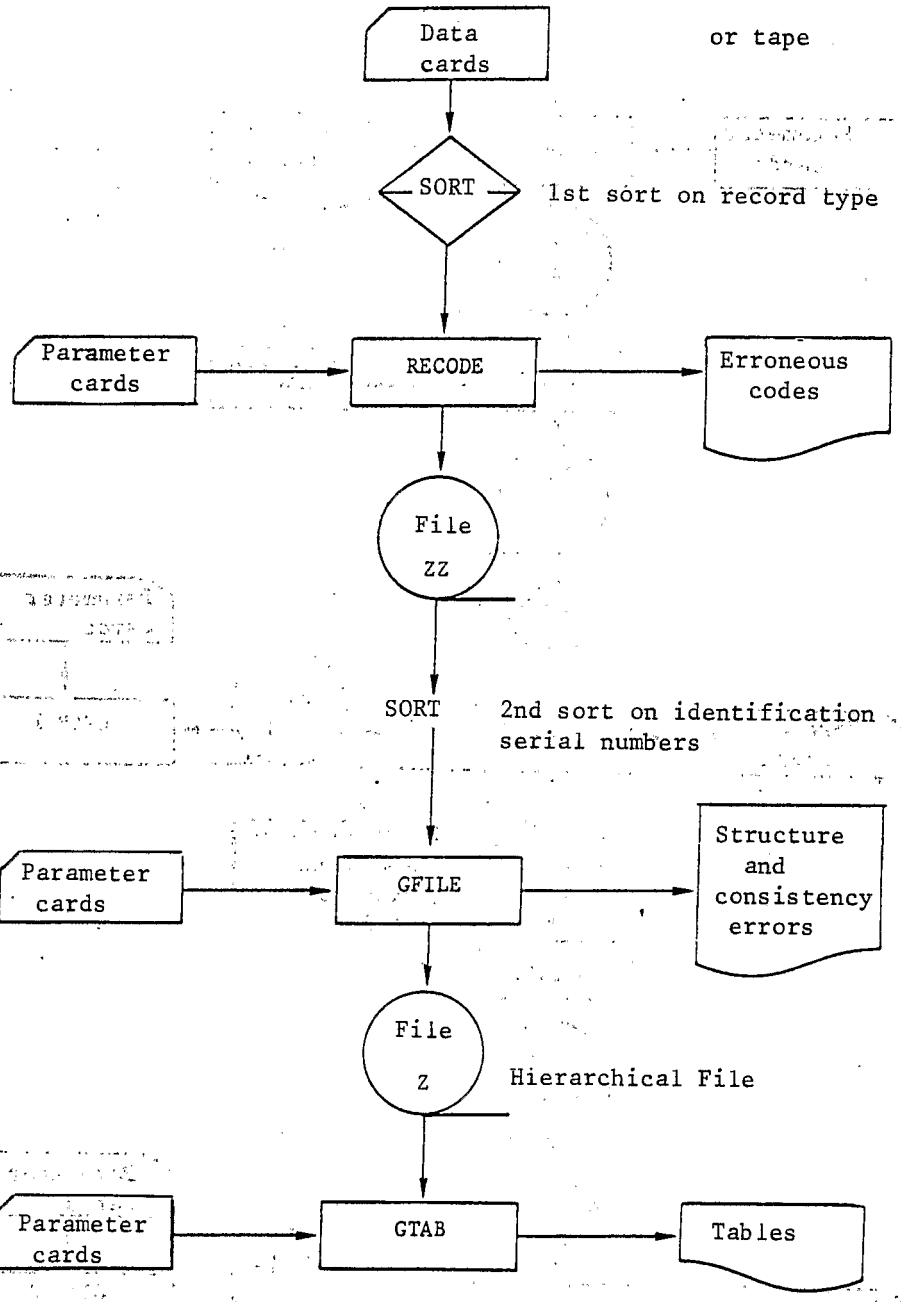
This program system allows broad possibilities of use, all operations being specified by parameter cards which are external to the programs (program data). Forms including definitions, abbreviated rules and boxes for the coding of the selected options are used as punched input documents for the production of the parameter cards. These forms are shown in Chapter 10. The name of the program system indicates its applicability in fields where synthesis between three hierarchies is desired; generally speaking to problems referring to mental, social and physical health. At the limit, the program is also applicable when data refer to one hierarchy only.

2. USAGE

2.1 Technical characteristics

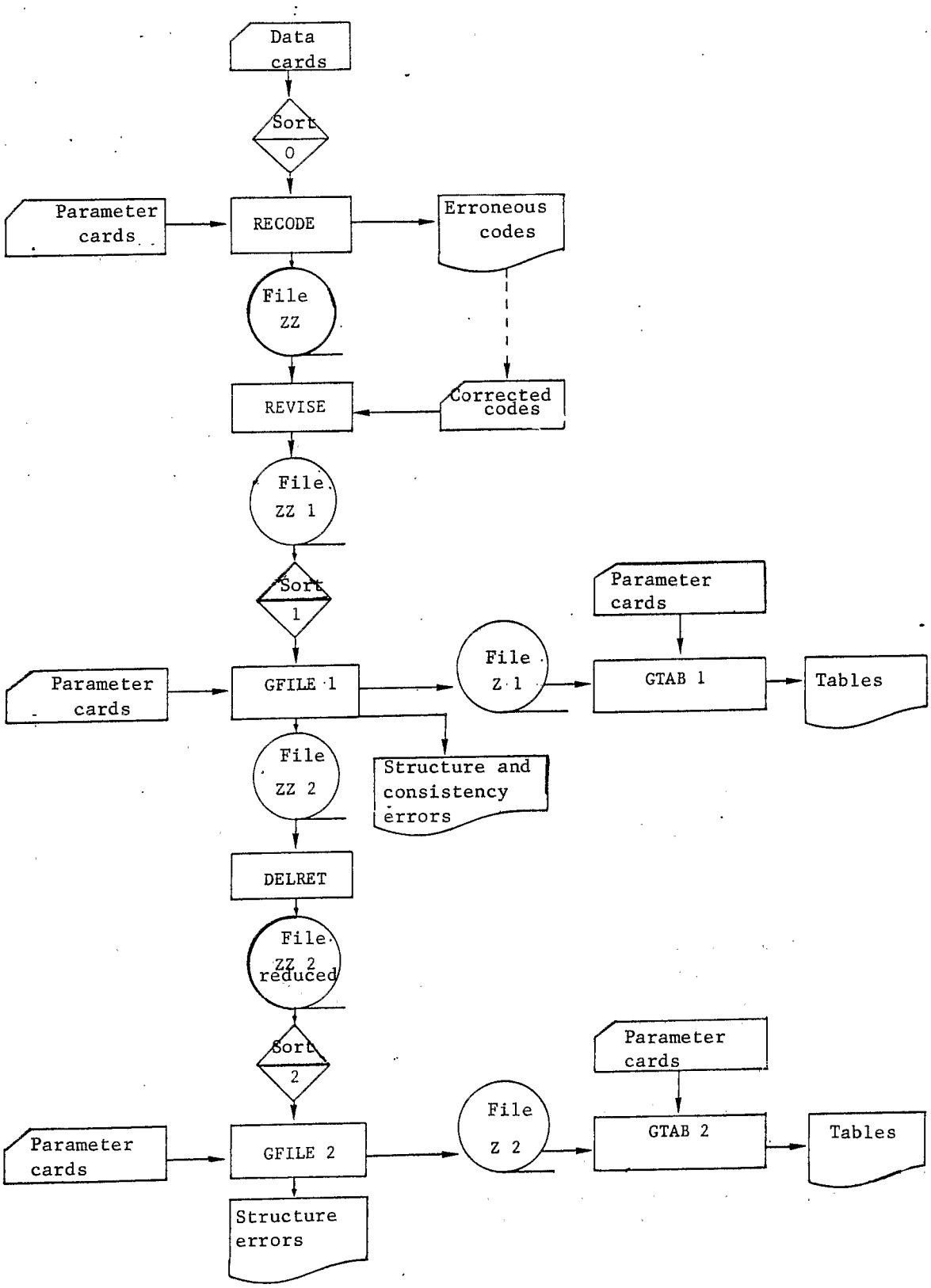
The programs of the TWH system have been written in Fortran IV G, using an IBM 360, 256 K, under OS (Operating System). At the programming level, optimization research was undertaken especially in the field of input output operations. The programs can be executed with a REGION parameter not bigger than 250 K, using tape units and a disk. The use of the programs

FIG. 1 GENERAL
FLOW CHART OF THE PROCESSING



Note: Error corrections are not inserted in this flow chart, but are inserted in figure 2.

FIG. 2. FLOW CHART OF A COMPLEX PROCESSING



requires the coding of parameter cards: on one hand these cards are input data to the TWH system programs, on the other hand these cards are used for single or sequential exploitation of these programs by the computer (JOB CONTROL LANGUAGE in OS system). The coding of these parameter cards is rather simple for current utilizations.

2.2 Analysis of the processing problem

The analysis of a processing problem can be undertaken before the design and coding of the questionnaire(s) or subsequently to these steps. In the first case the conception of the questionnaire system will be the last step in the analysis of the processing problem: the coding will be suitable for direct processing by TWH. In the second case the analysis will be performed as a function of the existing questionnaire(s) and coding; ad hoc editing programs will have to be written for the splitting or the concatenation of records or for the translation of alphabetic characters which cannot be read by RECODE. Manual recoding including code of format modification may also be a necessary preliminary step. The following steps are included in this analysis:

a) List of tables;

b) List of variables;

c) List of variables including categories, and category-groups;

d) The specification of the hierarchical files (Z) to be formed, which is the major step;

e) The flow chart of the processing.

2.3 The coding of parameter cards

The usage of the TWH system involves the systematic coding of all the questions which should be shown in the results of the survey. This coding, which will be sent to a punching unit, attributes to each question of the basic survey questionnaire a variable name (or question name) and at least one serial number. To each of these questions corresponds several possible answers (or measurements) with an old code (question) and a new code (for tabulation) and a name for the answer or the code. The variables for tabulation should also be entirely coded, with their name, their code names and their recoding parameters. This coding being done, a standard table referring to any of the coded variables can be obtained very easily. In this case only 2 to 8 serial numbers, referring to the variables in the table, will have to be punched. Other non standard tables can be obtained by the specification of one or more additional parameters. Other parameters correspond to table editing options or give access to specific tabulation techniques.

The coding is done on forms which are specifically designed for the system (chapter 9). Generally, the coding of numerical values is right adjusted in the defined field, the unused space of this field being left blank. The parameter cards are usually read as 'numerical' and blank is therefore interpreted as zero. However, in GTAB program, the reading of the parameter card C is partially alphabetical. In this manual, the questions and variables are identified by their Fortran name, this name being shown on the forms of chapter 10 and used in the programs. The field length for the coding of one parameter is either explicitly shown on the form, or otherwise covers 4 columns.

3. THE FUNCTIONS OF THE PROGRAMS

3.1 RECODE and REVISE

RECODE reads data from cards or tape (in character format, with or without internal blocking) and converts them to numerical integer data. As output, recoded data are written on tape or disk (file ZZ), one option allowing for the exclusion or the keeping of these

records containing one or several erroneous codes, i.e. which do not belong to the collection of selected codes which are specified in the parameter cards.

The reading is alphabetical and limited to 13 characters which are 0-9, & (code 11), - (code 12) and blank. RECODE allows for the transformation of each code of a variable of 1 or 2 characters into a numerical code of two digits (positive integer smaller than 100, or negative integer greater than -10). This program allows for several recodings of the same variable and for the elimination of information fields of the input records which will not be used for the analysis. It can function with two systems of subfiles, one being internal to the program, the other being connected with JCL. Parameter cards can be stored on a direct access file, with the view of repetitive and sequential processing of series of subfiles of various nature. For example, if yearly files or subfiles are produced, the yearly information referring to five different documents or records of different format, the subfiles will be grouped in the series 1-5, 6-10, 11-15,... for the years 1970, 1971, 1972 ....

The program edits error lists describing erroneous codes and records (with the old and new codes, according to the old and the new format). These lists can also be produced simultaneously on tape or punch cards.

The program REVISE allows for the correction of single codes. It can read the ZZ files from RECODE including the two subfiles systems. On option it works without control on identification sequence. This option is used for instance when corrections are done on identification variables.

3.2 GFILE

As input, GFILE reads a tape (file ZZ) which is the output of a SORT program. The SORT program groups the information relating to one individual, and the individuals which belong to a same unit (household for example). The program analyses these information groups and produces a hierarchical processing unit. It can then transfer information inside this processing unit, for example from the household record to the individual records and calculate new variables. The calculation of new variables is done with one or several relations, one parameter card being used for each relation, the number of relations inside the processing unit being limited to 300. A relation establishes a correspondance between three variables, the first two variables being used for reading, the third being a result for writing. These variables belong to units of information which might belong to the same or to different hierarchies. For instance, it is possible to select a relation between the individual record (mother, hierarchy 2) and its 'event' records (births, hierarchy 3) and report on the record 'mother' (with variable 3), the number of births specified as 'live birth, dead', which can be read either as a variable of the record 'birth' of hierarchy 3, or as a new variable of this same card, which was calculated for this purpose by means of one or several preceding relations.

Each relation uses one operation, this operation being a linear function, a rate, a test, a transfer, a count, or a conversion table. Other functions can be introduced in the program under the heading 'ad hoc operation'.

As output, the information is written on tape or disk, with hierarchical structure (file Z). It is also possible to have independently or simultaneously as output a tape (file ZZ) with the calculated variables. This latter output can be used as input to a GFILE with new specifications. The number of possible operations is thus unlimited. GFILE can read multiple files (subfiles) and can use the file (subfile) serial number for calculations. An option for the storage of the invariant parameters is available.

3.3 GTAB

GTAB reads as input a tape or a disk which contains the hierarchical file Z and edits tables, each of these being defined by two or four parameter cards. The standard table which is produced by the program is a frequency distribution with at most 8 variables and at most 24 000 cells, with marginal and subtable subtotals. GTAB produces on option tables with line

and column percentages, rates for questions including the three answers 'yes', 'no', 'unknown', elementary statistics such as the mean, the variance, the standard deviation, the correlation coefficient and related test. Subtotals and totals can be obtained for all the above defined tables. Parameters allow for various editing facilities or give access to various tabulation techniques. Thus GTAB can form various types of composed variables, referring to collections of variables with a similar structure. For example, the category 'list of variables' in the parameter 'type of variable' allows for the inclusion in a single table of at most 60 subtables of similar structure and relating to at most 68 variables. Other variables such as 'counting variables' or 'index variables' allow for the numerical description of certain characteristics of the processing unit or of its structure.

In addition GTAB produces tables which relate units which are located in different hierarchies, mainly with the view of producing rates and means. In this case it is possible to present after one single reading of the file, three classifications with similar structure each of these classifications including at most 8000 cells.

The input file is read as many times as the number of tables to be produced, these successive readings being supervised by the program.

One option allows for the reading of non structured files, the files being of fixed length format (VB or VBS), such as the output of RECODE, or the output of type ZZ from GFILE.

3.5 Optional programs

According to the amount and quality of the information, the use of certain auxiliary programs can facilitate the processing. Coding documents and dictionaries are produced by the programs which edit the parameter cards. These documents are very helpful at all stages of a study and for the creation of archives of a project.

3.5.1 Programs for the edition of parameter cards

3.5.1.1 LISCOD: List of old and corresponding new codes, old and new sequence of the variables.

3.5.1.2 LISVAR: Tabulation variables in storage sequence and tabulation variables in questionnaire sequence.

3.5.1.3 LISTAB: Tables in the sequence of the tabulation plan and tables according to tabulation variables.

3.5.2 Programs referring to the files

3.5.2.1 REVISE is used for the correction of erroneous codes on a file of type ZZ.

3.5.2.2 DELRET allows for the extraction from a ZZ file of units of information which satisfy conditions as defined by parameter cards, producing an abbreviated ZZ file. Elimination of erroneous records can also be obtained.

3.5.2.3 GPACK produces packed data, on the basis of RECODE specifications and punches the table of new addresses on punch cards. This table can be used as optional input to GTAB.

3.5.2.4 VARFIL is used for the storage, revision and extension of the file of tabulation variables.

3.5.2.5 ZZFILE prints the n first records of a ZZ file, according to a selected format.

3.5.2.6 ZTOZZ makes the conversion from a Z file into a ZZ file.

3.5.2.7 ZMAP prints the description of the general structure of the file. It prints the structure of the first n processing units. On option the complete content of the first n processing units is printed.

4. THE FILES

4.1 Three types of files

- Card or tape file: character information
- File ZZ: non hierarchical, integer information
- File Z: hierarchical, integer information

Integer information generally occupies four bytes. On option the four byte words can contain several variables. In this latter case the record length will be reduced, the time taken by input-output operations will be reduced, the calculating time will however be increased (Program GPACK).

4.2 Example (see figure 3)

In a survey information is collected from households, individuals within these households and events which had occurred to these individuals.

The information is of hierarchical nature and can be described as a tree diagram.

```
                    ( individual 1        ( events A
                    (                     ( events B
                    (
Household           ( individual 2        ( events A
                    (                     ( events B
                    (
                    ( individual 3        ( events A
                    (                     ( events B
```

In the initial file, these data will be punched on cards, for example with:

2 cards describing the household
1 card for each individual
1 card for each event (A and B)

In RECODE, each category of each above record will be treated separately as a unit of information. The file Z (and ZZ) will include 5 categories of units of information (2 for the households, 1 for the individuals, 1 for the events of type A, 1 for the events of type B). The collection of units of information referring to the same household forms one element of the file Z which is the processing unit for the calculations to be done by GFILE. Rules concerning the processing of an element of Z are described in 6.2 (relations between variables, GFILE) and in 7 (classification and reading of several categories, GTAB).

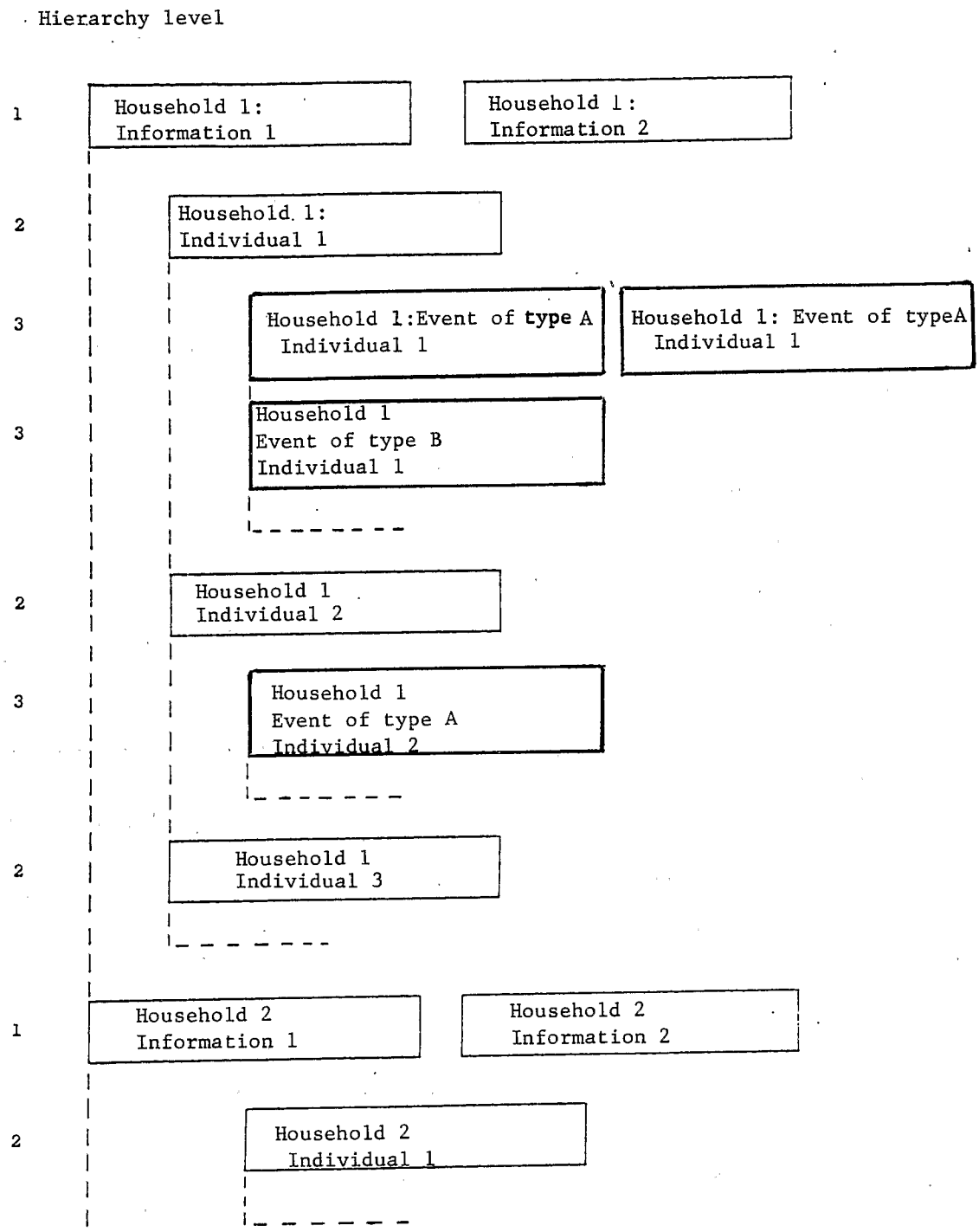
4.3 File Z (hierarchical)

This file can have 3 levels:

Level 1: Constant number (0-20) of categories of units of information each containing 1 single unit of information.

Level 2: Variable number of units of information, but 1 single category of units of information.

FIG. 3. Example of hierarchical file



Level 3: Constant or variable number of categories of units of information.

For the description of the file, the units of information are classified into categories. Three ways of numbering these categories are used:

- for all categories, sequentially, starting at level 1
- for the categories of levels 2 and 3, sequentially, starting at level 2
- for the categories of level 3, sequentially, starting at level 3

NUMBERING OF CATEGORIES

(Example of figure 3)

Level	Category of unit of information	First sequence number	Second sequence number	Third sequence number
1	Household: information 1	1	-	-
1	Household: information 2	2	-	-
2	Individual	3	1	-
3	Event type A	4	2	1
3	Event type B	5	3	2

The categories of units of information on the first level can be identified by only the first sequence number.

The categories on level 2 can be identified by the first and second sequence number.

The categories on level 3 are identified by 3 sequence numbers.

At level 2, there is only one category of units of information, there is therefore a difference of 1 between the second and the third sequence number.

The first and the second sequence numbers are confounded when the number of categories of units of information at level 1 is zero.

For the levels 2 and 3, the number of units of information of each category does not influence the sequence numbers.

4.4 The controls

The errors to be detected can be classified into three groups:

- error on a single variable, when this variable takes a value which is different from the predicted values

- error on crossed variables, when the value taken by a variable restricts the field of possible values for another variable; (or the values taken by several variables restrict the field of possible values for another variable)

- structure error, when a number of units of information on level 1, 2 or 3 differs from its specified value.

With RECODE only errors on single variables are treated (see 5.4).

With GFILE, the file description leads to a control on the file structure (see 6.1.4.5).

With GFILE, the relations of type 'test' allow for the control of errors on single and crossed variables (see 6.2.4.1).

5. THE RECODING PROGRAM RECODE

5.0 Job description (Form B, card 1)

TITLE: Job title (at most 68 characters)

MODCOR: Printing of the instructions for the revision (see 5.6)

INT: Code 0: no temporary storage

Code 13: temporary storage of the parameter cards from RECODE (Forms C, D, E, F, (G), (A))

NFDIR: Number of categories of units of information which go on temporary storage.

5.1 Input file description (Form B, card 2.1)

IN: Fortran data set reference number for the input unit

NBLIN: Number of units of information per record

LBLIN: Number of characters per unit of information

For a card file: NBLIN x LBLIN should be smaller or equal to 80.

Example 1 NBLIN = 1, LBLIN = 65

Example 2 NBLIN = 2, LBLIN = 35: if the field extending from column 36 to column 70 is blank, it will be considered as a unit of information, and the rejection of this record has to be specified in the appropriate parameter cards for RECODE.

For a tape file: LBLIN should be smaller or equal to 1000.

Remark: If a card file contains more than 1 card per unit of information it is possible either to create a tape file by an ad hoc program which matches the cards, or to accept that each card is a new unit of information. This latter alternative will affect the file structure (see 6.4.1 (GFILE), 7(GTAB)).

5.2 Description of output file ZZ (Form B, card 2.1)

IUT: Fortran data set reference number for the output unit

NBL: Number of units of information per record

NEC: Number of words (4 bytes) per units of information

MLRECL: Code 1 for records of 1820 words

Code 2 for records of NBL * NEC words

If the file has to be sorted specify NBL = 1 and MLRECL = 2.

5.3 Identification of the variables

Each variable is identified by a serial number.

5.3.1 Input file (Form A)

One variable has 1 or 2 or (3 or more) characters. Dummy variables are used to skip input fields of specified length. This field length is measured in number of characters and can be positive or negative.

Example:

Variable (pseudo variable) serial number	Number of columns (characters) covered	Column allocation
1	2	1-2
2	1	3
3	4	4-7
4	Skip = 5	
5	2	13-14
6	Skip = -2	
7	2	13-14

In this example, the variable 7 is a new codification of columns 13-14. Each variable is thus characterized by its serial number and the field covered (number of columns). The position of each variable is determined by the number of characters taken by the variables and pseudo variables with a lower serial number in the original sequence.

5.3.2 Output file (Form G)

The output file contains only the variables. The intermediate serial number of the variable results from the suppression of the pseudo variables.

Example (continued):

Initial serial number	Intermediate serial number	Final serial number
1	1	1
2	2	3
3	3	4
5	4	2
7	5	5

The serial number can on option be modified according to specifications given in Form G: in this case the final serial number differs from the intermediate serial number (NEWCO = 1, form A). In above example Form G will contain successively 1, 4, 2, 3, 5 (intermediate serial numbers in the final sequence).

If NEWCO = 0, the final and intermediate serial numbers are equal.

5.4 Processing of the variables

5.4.1 Variables of 1 character

The number of these variables (NV1) is specified in form B, card 2, it is less than or equal to 630. For each variable, one card should be punched according to Form C. This card contains the variable serial number (initial) and gives for each punch 0 - 99, &, - and blank the new integer codes for the variable (ranging from -9 to 99). If, for a given punch no new value is specified this new value is zero. This card also contains a field for the specification of the error character (blank and a value in the range -9 to 99 can be selected as error character). On this card, this selected error character should be taken as the new code for each unacceptable original code.

5.4.2 Variables of 2 characters

The number of these variables is specified by NV2 (Form B, card 2.1), it should not be greater than 40. For each variable, 3 cards should be punched according to Form D.

NN is the serial number of the variable (initial). A new code value (from -9 to 99) is given for each old code value (0 - 99, &, -, blank) in a table of 103 values.

If NOCOL = 2 and if a new code value is blank or zero this new code value is made equal to the corresponding old code value: i.e. this code is not modified. This option is used when most of the code values remain unchanged and when none of the codes 1-99 should be replaced by zero. If NOCOL = 12 and if a new code value is blank or zero this new code value is now zero. This option is used when one of the old codes 1 - 99 should be replaced by zero. With this option, the writing of 103 new codes cannot be avoided. The characters are examined from right to left: if the right most character is blank, & or -, the following character is not examined. An error character should be selected as described for variables of one character (5.4.1).

5.4.3 Variables of 3 or more characters

For each of the NV3 variables (Form B, card 2.1), one card is punched according to Form E. The maximum number of these variables is 20. Each card gives the serial number of the variable (initial) and the number of characters of the variable (3 or more). Old code values are kept. A new value can be specified for blank, & and -. In addition to the error character a lower and an upper limit of the range of acceptable old code values can be specified as well as two values outside this range.

5.4.4 Dummy variables

The number of these variables is NS (Form B, card 2.1) and should not be greater than 20. One card is punched for each dummy variable according to Form F. This card gives the variable serial number (initial) and the number of characters covered.

5.5 Processing of erroneous records (Form B, card 2.2)

The units of information containing 1 or several erroneous codes can be included in or excluded from the output file ZZ (Fortran data set reference number IUT).

If INUTER = 1 no rejection of erroneous records.
If INUTER = 2 rejection of erroneous records.

This option is however only possible with the parameter MODCOR2 = 2 (Form B, card 2.1), i.e. when error messages refer to erroneous codes. In fact the processing can be done in two ways.

5.5.1 For small files, or initial tests on quality of data

In form B, cards 2.1 and 2.2:

MODCOR2 = 1 (col. 77-80)

IER1 = 0
IER2 = 0

For each erroneous unit of information the following is printed:

a) the new variables, in intermediate sequence (5.3.2), erroneous variables being indicated by stars

b) the unit of information in character format (original format)

Remark: With this option erroneous units of information are excluded from the output file ZZ.

5.5.2 For final production

MODCOR2 = 2 (col 77-80, Form B, card 2.2): editing specifications for the error lists are specified in Form B2. Only the erroneous variables are printed.

NVAID = number of variables reserved for the identification of the unit of information. (Maximum = 10)

NIDRC Intermediate serial number (5.3.2) of each of the NVAID identification variables.

IDRC Text (80 characters at most) describing these NVAID parameters which are selected for the identification of one unit of information of the ZZ file.

IDVA Text (80 characters at most) describing the parameters which identify one variable within one unit of information.

Example: (see list of erroneous variables, figure 4)

IDVA is the text following 'VARIABLE': i.e. card column, item.

IDRC is the text following 'ENREGISTREMENT - RECORD': i.e. Patient serial number, punch card serial number. The first identification variable is the serial number of the patient and the second is the card category (punch card number).

The first erroneous variable is in column 35, its name follows, it is the 24th variable in the intermediate sequence. The first identification variable is equal to 9 and the second equal to 1. The third erroneous variable refers to the same unit of information as the second erroneous variable. Therefore the serial number of the third erroneous variable (70) is followed by 1. The descriptions of the variables (for example: 6-7 Date of birth) are written on Form A in columns 1-28, column 29-30 indicating the number of characters covered, the columns 31-33 indicating the initial serial number of the variable, the columns 34-36 indicating the intermediate serial number, the columns 37-39 being left blank if the final serial number does not differ from the intermediate serial number.

Two other options are possible:

a) The units of information are written in original format (character) on a file, the Fortran data set reference number of which is IER2; if IER2 = 0, this file is not created. If IER2 is not zero this file is printed at the end of the processing of one category of information.

b) The erroneous units of information are written with the new codes, the sequence of the variables being the final one: this is written on a file, the Fortran data set reference number of which is IER1; if IER1 = 0 this file is not created.

MODCOR2 greater than 2 (col 77-80, Form B, card 2.2): an additional error list is produced to the lists given by the option MODCOR2 = 2. MODCOR2 is in this case interpreted as the Fortran data set reference number for a file (cards or tape) containing the numerical identifications of the erroneous variables but excluding the alphabetical identifications. A record of this file includes:

the serial number of the error;

the serial number of the erroneous variable;

the index NSCA	code 0 or blank	new record
	code 1	same record

ID: The intermediate serial numbers of the variables which are used as identifiers.

The format of this record is (I6, I5, I1, 10I6). The file of erroneous variables can be read by REVISE. The correction in old code should be written right adjusted in the field 73-78 and will be read with the format 6A1. The correction in new code is read with I6 format, it should be right adjusted in the field of 6 characters which follows the field reserved for the identifier with the ultimate rank NEW (see 8.2).

5.6 Correction mode (Form B)

MODCOR1 = 2 allows for the printing of a text (46 punch cards) giving the instructions for the coding of corrections (Form B, card 1).

5.6.1 Correction of the whole unit of information

In this case erroneous units should be eliminated from the output file ZZ of RECODE (INUTER = 2). The corrected units of information must be processed once more by RECODE.

FIG.4 LISTE DE CODES A REVISER-LIST FOR CODE REVISION

CARTE-CARD 1

VARIABLE: CARD COLUMN,NAME OF ITEM
ENREGISTREMENT-RECORD: PATIENT SERIAL NUMBER,PUNCH CARD NUMBER

ITEM	VARIABLE		ENREGISTREMENT-RECORD	REPONSE ANSWER
1	35	DAY COUGH WINTER III	24 ID100009ID200001ID300000ID400000	RA
2	6-7	DATE OF BIRTH	3 ID100030ID200001ID300000ID400000	RA
3	6-7	DATE OF BIRTH	701ID100030ID200001ID300000ID400000	RA
4	23	DURATION OF FOLLOW-UP	14 ID100045ID200001ID300000ID400000	RA
5	23	DURATION OF FOLLOW-UP	14 ID100056ID200001ID300000ID400000	RA
6	17	CHANGE OF WEIGHT O/O	10 ID100198ID200001ID300000ID400000	RA
7	6-7	DATE OF BIRTH	3 ID100258ID200001ID300000ID400000	RA
8	6-7	DATE OF BIRTH	701ID100258ID200001ID300000ID400000	RA
9	74	DYSPNOEA 3, III	63 ID100317ID200001ID300000ID400000	RA
10	6-7	DATE OF BIRTH	3 ID100417ID200001ID300000ID400000	RA
11	6-7	DATE OF BIRTH	701ID100417ID200001ID300000ID400000	RA
12	24-25	START OF FOLLOW-UP	15 ID100420ID200001ID300000ID400000	RA
13	26-27	END OF FOLLOW-UP	161ID100420ID200001ID300000ID400000	RA
14	36	PERIOD AND CHANGE	251ID100420ID200001ID300000ID400000	RA
15	41	CHRONIC COUGH I	301ID100420ID200001ID300000ID400000	RA
16	42	CHRONIC COUGH II	311ID100420ID200001ID300000ID400000	RA
17	44	PERIOD AND CHANGE	331ID100420ID200001ID300000ID400000	RA
18	48	PERIOD AND CHANGE	371ID100420ID200001ID300000ID400000	RA
19	51	DAY PHLEGM WINTER III	401ID100420ID200001ID300000ID400000	RA
20	52	PERIOD AND CHANGE	411ID100420ID200001ID300000ID400000	RA
21	53	PHLEGM IN SUMMER I	421ID100420ID200001ID300000ID400000	RA
22	54	PHLEGM IN SUMMER II	431ID100420ID200001ID300000ID400000	RA
23	55	PHLEGM IN SUMMER III	441ID100420ID200001ID300000ID400000	RA
24	56	PERIOD AND CHANGE	451ID100420ID200001ID300000ID400000	RA
25	60	PERIOD AND CHANGE	491ID100420ID200001ID300000ID400000	RA
26	63	HAEMOPTYSIS	521ID100420ID200001ID300000ID400000	RA
27	64	DYSPNOEA 1,I	531ID100420ID200001ID300000ID400000	RA
28	68	DYSPNOEA 2,I	571ID100420ID200001ID300000ID400000	RA
29	69	DYSPNOEA 2,II	581ID100420ID200001ID300000ID400000	RA
30	70	DYSPNOEA 2,III	591ID100420ID200001ID300000ID400000	RA
31	71	PERIOD AND CHANGE	601ID100420ID200001ID300000ID400000	RA
32	75	PERIOD AND CHANGE	641ID100420ID200001ID300000ID400000	RA
33	33	DAY COUGH WINTER I	22 ID100442ID200001ID300000ID400000	RA

FIN-END,NOMBRE D'ENREGISTREMENTS RECODES-NUMBER OF RECODED RECORDS= 445

See explanations in 5.5.2.

The resulting new ZZ file must be concatened with the preceding one or must be written at the end of the preceding one, using the JCL parameter DISP = (,MOD).

5.6.2 Correction of erroneous codes: use of REVISE

Corrections are punched from the list 5.5.2 (figure 4). These corrections can be reported either on the major ZZ file (if INUTER = 1), or on the ZZ file which is described in 5.5.2b (if INUTER = 2 and IER1 is not zero). In this latter alternative the file of corrected units must be concatenated with the error free ZZ file.

5.7 Organization of the parameter cards

5.7.1 Either a single category of units of information is treated or several. In the former alternative, there is only one set of parameter cards to be prepared, in the latter as many sets as categories of units of information are needed (these categories corresponding generally to various card designs). However several categories can also be created when different treatments are applied on cards with identical design.

Consequently the number of parameter card sets varies not only with the number of categories but also when subfiles are created. The possibility of creating a multiple file gives the opportunity of preparing in a single job subfiles referring for example to various periods or different geographical survey areas.

The end of a subfile is tested on the value of one variable (with intermediate serial number KEND) and the tested value is NEND (new code). The parameter card sets consist of the following elements (parenthesis (-) indicate optional cards).

Cards A

- FORM B, card 1 Job title

  MODCOR1 (see 5.6)

  INT = 13 (INT = blank if no temporary storage)

(-) 46 cards if MODCOR1 = 2

Cards B

- FORM B, cards 2.1 and 2.2 Input file (see 5.1)

  Output file (see 5.2)

  Number of variables by field length (see 5.4)

  NSPEC (see 5.7.2, 5.7.3, 5.7.4 and 5.7.5)

  KEND Intermediate sequence number of a variable on which the new code value NEND is tested, a positive test result will be followed by a modification of a category of units of information and/or a subfile (see below)

  NRUNS = 1 except for the last run of the job, for which NRUNS = 0

  NPRINT = 1. Output records are printed;

  = 0 the last output record only is printed

LEND Usually coded 9999. If coded 0 it is considered as end of subfile by GFILE (see 5.7.2)

NDFIL (see below)

NEWCO = 0 if the final serial number of the variable is equal to the intermediate serial number

= 1 if these numbers are different (see 5.3.2)

IN, IUT, IER1 and IER2 should be different from 3 and 13, these Fortran data set reference numbers being respectively reserved for the SYSOUT unit and the direct access unit for the temporary storage of the parameter cards. The direct access file is defined by DEFINE FILE 13 (60, 906, U, KK).

Cards C

(-) FORM B2 : 3 cards, if MODCOR2 = 2

Cards D

- FORM C NV1 parameter cards for variables of 1 character (see 5.4.1)

- FORM D 3xNV2 parameter cards for variables of 2 characters (see 5.4.2)

- FORM E NV3 parameter cards for variables of 3 characters or more (see 5.4.3)

- FORM F NS parameter cards for the skips (see 5.4.4)

(-) FORM G (NEC-1)/20+1 parameter cards for the modification of the variable serial number, when NEWCO is positive (see 5.3.2)

(-) FORM A Name of the variables, if MODCOR2 = 2 (exclude skips, include new variables, and keep the reading sequence).

5.7.2 A single category of units of information - Simple file

Sets of cards A B (C) D

Specificities for -A : NFDIR = 0

-B : NSPEC = 1

KEND any variable serial number (intermediate serial number)

NEND ad hoc new code value

Usually special system cards are introduced in the input file for RECODE (1 for each category or subfile) which contain these special values for NEND. These sytem cards do not appear in the output ZZ file. However a record with the value LEND for each variable is printed at the end of a ZZ subfile or category.

NRUNS = 0

NDFIL = 1

5.7.3 A single category of units of information - Multiple file

Sets of cards A B1 (C1) D

B2 (C2)

.....

Bm (Cm)

m being the number of subfiles

Cards A and D are inserted only once.

Specificities for -A : NFDIR = 0

-B : NSPEC = 1 for B1

= 0 for B2, ..., Bm

NRUNS = 1 for B1, ..., Bm-1

= 0 for Bm

NDFIL = 1

KEND, intermediate serial number of the variable on which the end of the subfile is tested (the test value is the new code for NEND) (see also 5.7.2). The value NEND must be found exactly m times for the complete processing of the file.

5.7.4 Several categories of information - Simple file

Sets of cards A B1 (C1) D1

B2 (C2) D2

.....

Bn (Cn) Dn

n being the number of categories of units of information.

Specificities for -A : NFDIR = 0

-B : NSPEC = 1 (the serial number of the category would also be accepted)

KEND, intermediate serial number of the variable on which the end of a category is tested (the test value is the new code for NEND) (see also 5.7.2). The value NEND must be found exactly n times for the complete processing of the file.

NRUNS = 1 for B1, ..., Bn-1

= 0 for Bn

NDFIL = 0 for B1, ..., Bn-1

= 1 for Bn

5.7.5 Several categories of units of information - Multiple file

```
Sets of cards : A          )
                           )
                B 0,1 D1   )
                           )
                B 0,2 D2   )    Disk storage
                           )
                ........   )
                           )
                B 0,n Dn   )

                B 1,1 (C11) )
                            )
                ........... )  Subfile 1
                            )
                B 1,n (C1n) )

                ...............................................................

                B m,1 (Cm1) )
                            )
                ........... )  Subfile m
                            )
                B m,n (Cmn) )
```

m being the number of subfiles, n the number of categories.

Specificities for -A : NFDIR = number of categories of units of information (n)

-B : NSPEC = First sequence number of the category (1-n)

KEND, intermediate serial number of the variable on which the end of a category in each subfile is tested (the test value is the new code for NEND) (see also 5.7.2). The value NEND must be found exactly m x n times for the complete processing of the file.

NRUNS = 1 for all B except B m,n

= 0 for B m,n

NDFIL = 0 for B i,1; ...; Bi, n-1

(i = 0 to m)

= 1 for B 0,n; ...; B m,n

6. THE PROGRAM FOR FILE CREATION GFILE

6.1 The files (see 4)

GFILE reads an input file of type ZZ. As output a selection among 3 options is possible.

NEWFIL = 0 a file of type Z (FORM A)

NEWFIL = 2 a file of type ZZ

NEWFIL = 1 the 2 files Z and ZZ

6.1.1 The input file (FORM A)

IN = Fortran data set reference number

NBL = Number of units of information per record

LSBL = Number of words per unit of information

IFPACK = 1: Packed data : several variables in 1 word

= 0: 1 variable per word : this is the case when the file is the output of RECODE

KTYPO = Intermediate serial number of the variable containing the code of the category of the unit of information.

KHSPO = Intermediate serial number of the variable containing the identification of these units of information which form one processing unit (example: if the processing unit is a household the variable with the intermediate serial number KHSPO of each unit in this household contains the household serial number)

IWX = Number of words in the Z or ZZ zone (see below in 6.1.4.3).

6.1.2 The output file ZZ (FORM AA)

IUTZZ = Fortran data set reference number

NBLZZ = Number of units of information per record (NBLZZ = 1 if this file must be sorted)

LSBLZZ = Number of words per unit of information

Input and output files have the same length if LSBLZZ = LSBL + IWT-IWX. If the zone content has to be maintained IWT = IWX = 0.

NTYPZZ and KTYPZZ allows for selections on the category of units of information.

NTYPZZ = Number of categories of units of information which are selected for ZZ.

KTYPZZ = First sequence number of each selected category.

6.1.3 The output file Z (FORM B, card 4)

IUT = Fortran data set reference number.

MWB = 1820 (number of words per record)

IWT = Number of words inserted in the Z zone (see 6.1.4.3)

KPO = For each of the IWT words of the Z zone, serial number (final) of the variable on the X or ZZ file, X being the input file to GFILE or the output file of RECODE.

6.1.4 File structure

6.1.4.1 Sorting

All units of information belonging to the same processing unit have one identification variable containing the same value, this variable being the principal identification variable in Z (for example: the unique household serial number). The file must be sorted according to this variable (usually taken as the major sorting variable) which has the final serial number KHSPO on all units. In the processing unit the units must be classified in the

following sequence: units of level 1, first unit of level 2 and related units of level 3, second unit of level 2 and related units of level 3, ... This sequence is introduced when the sorting variables which follow the principal identification variable are, for example, the individual serial number (in preselected fixed position) and the category serial number (in position KTYPO). If, for example, the household serial number is not unique, the sort will be performed on a more general primary identity, like the village, and then on the household serial number (position KHSPO).

For a file with 3 levels and several units on level 2 the three above mentioned identification variables are compulsory. Other sort variables might be necessary for the classification of level 3 units, according to the extension of the file structure (or structures). All identification variables constitute the input zone X or ZZ (see 6.1.4.3). All units of the input file ZZ have the same zone. A zone can contain, as exceptions, variables with the same serial number but having different meaning in the different units. These variables will not be appropriate for sorting.

6.1.4.2 Number of units of information

(FORM A) NCH : Number of categories of units of information on level 1 (0 to 20)

NTYP : Number of categories of units of information on level 3 (0 to 20)

NIFIX : Number of units of information on level 2 when constant, or zero when variable

NIMAX : Estimated maximal number of units of information on level 2, when this number is variable. If unknown: code 0. If constant: code NIFIX. Option 0 requires longest execution time. NIMAX should not exceed the value 200/(NTYP + 1)

(FORM B, card 3) LSEQ: Number of units of information for each of the NTYP categories. (0 = undetermined)

6.1.4.3 Common zone

The common zone contains the IWX first words of each unit of information, these words being generally identification variables. This zone is eliminated from file Z and cannot be used by GFILE unless the option to keep IWT words is selected.

Example IWX = 20
IWT = 2, it is desirable to keep the 5th and the 12th zone word in the Z file. KPO is then coded as 5 and 12. These two words occupy the positions 3 and 4 in the processing unit. The common zone is maintained on the output file ZZ and is recopied from the first unit of information of the processing unit, the content of the common zone remaining therefore the same for all units of the same processing unit. If the content of the zone of each unit has to be maintained unchanged the options IWX = 0 and IWT = 0 should be selected.

6.1.4.4 Number of words per unit of information

The input file ZZ contains fixed length records, LSBL being selected as the maximum number of words referring to any unit of information, taking account of sorting operations which might be incompatible with variable length records. As output, this maximal length (LSBLZZ) can be greater or smaller than LSBL. This allows for the suppression of the last variables or for the addition of new variables. In the latter case the new variables are initialized with the value zero.

The file Z :

(FORM A)	NWI	:	Number of words in the units of information on level 2 (excluding the IWX words of the common zone).
(FORM B, card 1)	NWH	:	Number of words in the units of information on level 1 (excluding the IWX words of the common zone) for each of the MCH categories.
(FORM B, card 2)	NWE	:	Number of words in the units of information on level 3 (excluding the IWX words of the common zone) for each of the NTYP categories.

The output file ZZ has fixed length records, the record length being the product of the subblock length in words by the number of subblocks, i.e. LNBLZZ = LSBLZZ x NBLZZ (see 6.1.2).

6.1.4.5 Controls of structure

The controls of structure concern the number of units of information in the various categories (see 4.3 and 6.1.4.2). If a structure error is identified, an error message is edited, (see fig. 5 and 6.2.4.1), and the units which do not form a processing unit are rejected into the file of erroneous processing units. The following checks are done:

The number of units on the first level must be exactly NCH

If NIFIX is not zero the number of units on level 2 are exactly NIFIX. The unit of rank NCH + 1 should be on level 2.

If LSEQ is not zero for certain categories of units, there should be exactly LSEQ (K) units of information on level 3 in this category of units for which the third sequence number is K.

The units of level 3 which refer to the same unit of level 2 must be in the sequence given by the file structure (KTYPO, LSEQ). Only when units of level 3 are not represented or not existing can 2 units of level 2 be consecutive.

6.2 Processing the relations

6.2.1 GFILE has two main functions, the first being the formation of a structured file (see 6.1.4) and the second (optional) being the performance of consistency tests between variables and the calculation of intermediate and new variables, within or between the units of the same processing unit.

To each operation (test or calculation) corresponds one or several relations. The relations are performed in their writing sequence, their maximal number is 300. These relations can call for tables, functions or tests which are limited by the following parameters:

NREL	(FORM B, card 4)	Number of relations	(0 - 300)
NNEWV	(FORM B, card 4)	Number of tables	(0 - 10)
NNEWF	(FORM B, card 4)	Number of functions	(0 - 200)

The relations are described on FORM C.

6.2.2 Possible relations

The relations are only possible within one processing unit of the file Z. Moreover any two units of information cannot always be related directly. On the tree diagram described in 4.2 variables can be related when they belong

- to events of type A referring to one individual
- to events of type B referring to one individual

FIG. 5 GFILE: LIST OF ERROR MESSAGES

STRUCTURE ERROR	1	4	42	991	6	29	1	2	1	22							
STRUCTURE ERROR	1	4	43	1818	5	29	1	2	1	29							
STRUCTURE ERROR	1	4	47	2353	5	12	1	2	1	31							
STRUCTURE ERROR	1	4	46	2723	5	2	1	2	1	26							
STRUCTURE ERROR	1	4	46	2724	5	19	1	2	1	20							
IMPLICATION ERROR	1	4	47	3805	4	14	1	2	1	25	16	4	1	0	1666	1	1
STRUCTURE ERROR	1	4	42	4221	3	12	1	2	1	30							
IMPLICATION ERROR	1	4	47	5019	3	22	1	2	1	23	16	4	5	4	1666	1	1
IMPLICATION ERROR	1	4	43	6132	2	19	1	2	1	23	16	4	5	3	1014	1	1
STRUCTURE ERROR	1	4	43	6202	2	10	1	2	1	30							
STRUCTURE ERROR	1	4	43	6203	2	12	1	2	1	23							
STRUCTURE ERROR	1	4	43	6204	2	2	1	2	1	49							
STRUCTURE ERROR	1	4	43	6205	2	6	1	2	1	26							
STRUCTURE ERROR	1	4	43	6206	2	24	1	2	1	28							
IMPLICATION ERROR	1	4	44	6985	2	27	1	2	1	27	16	4	3	2	362	1	1
STRUCTURE ERROR	1	4	44	7295	2	27	1	2	1	25							
STRUCTURE ERROR	1	4	41	7749	1	17	1	2	1	23							
STRUCTURE ERROR	9999	9999	9999	9999	9999	9999	9999	9999	9999	9999							

See explanations in 6.1.4.5 and 6.2.4.1

- to events of type A referring to one individual and variables belonging to this individual and to the household

- to events of type B referring to one individual and to the household

- to the individuals and the household

However no relation can be established between variables belonging to

- events of type A and events of type B referring to the same individual (unless the events Ai, Bi form couples)

- events of type A referring to different individuals

- events of type B referring to different individuals

- events of type A or B referring to one individual and variables from another individual

- two different individuals

In other words only these units of information belonging to the same branch can be directly related, i.e. if two units must be related, they should not be on two parallel branches.

For the establishment of relations between events of type A and B of the same individual these events can be considered as belonging to the same category (another variable being taken to define subcategories), or they can be coupled; or the relation is divided into two parts, the intermediate term of the comparison or result being stored on the individual (level 2 unit).

6.2.3 Type of relations: KREL

The type of relation is described by 3 indices I, J, K.

- I indicates the levels which are related, its value ranges from 1-12 depending on the levels.

	Zone	Level 1	Level 2	Level 3
Zone		7	8	9
Level 1		1	11	(12)
Level 2			2	4, 6, 10
Level 3				3, 5

I = 12 is in fact a relation between levels 1, 2 and 3.

- J indicates, for those relations which are involved, if the units of information are taken in the progressive sequence (J=0), or in the regressive one (J=1). The sequence results from the sorting of the input file ZZ.

- K indicates, for those relations inside level 3, if the consecutive units are 1 and 2, and then 2 and 3, ..., in this case K = 1; or if the pairs of unit are 1 and 2, then 3 and 4, ..., in this case K = 2. For all other relation K = 0.

KREL is the combination of I, J and K:
KREL = 100 x I + 10 x J + K.

6.2.3 List and characteristics of the relations

A relation contains generally 3 words, the third word being an arithmetical function of the first two. The various operations are described in 6.2.4. For certain relations the words cannot be selected freely in any unit, this being indicated below when appropriate (I = 5, 7, 8, 9). The list of relations follows:

I = 1 within and between units of level 1: J = 0, K = 0

I = 2 within each unit of level 2: J = 0, K = 0

I = 3 within each unit of level 3 of a given category: J = 0, K = 0

I = 4 between a unit of level 2 and the units of level 3 of a given category:
J = 0 or 1, K = 0

I = 5 between units of level 3 of a given category: J = 0 or 1, K = 1 or 2
1st word in the 1st unit
2nd and 3rd word in the second unit

I = 6 between a unit of level 2 and the first (J = 0) or the last (J = 1) unit of level 3 of a given category: J = 0 or 1, K = 0

I = 7 between the zone and the units of level 1: J = 0, K = 0
1st word in the zone (KCAR1 = 0)
2nd word in the level 1

I = 8 between the zone and all units of level 2: J = 0, K = 0
1st word in the zone (KCAR1 = 0)
2nd word in the level 2

I = 9 between the zone and all units of level 3 of a given category: J = 0, K = 0
1st word in the zone (KCAR1 = 0)
2nd word in the level 3

I = 10 double relation between a unit of level 2 and all units of level 3 of a given category: 1st relation I = 10, J = 0 or 1, K = 0. 2nd relation I = J = K = 0. If J of the first relation = 0, the first relation and the following one are executed between data referring to the individual and the first unit of level 3 of the selected category, then a similar operation including now the second unit of level 3 follows, ...

I = 11 between a unit of level 1 and all units of level 2: J = 0, K = 0

I = 12 between a unit of level 1, all units of level 2, and for each unit of level 2 all units of level 3 of a given category: J = 0, K = 0

6.2.4 Operations

An operation is described primarily by the parameter IREL, the value of which calls either for a table, a function or a test value. An operation is done usually with three variables, called W1, W2, W3; these variables being specified by the following parameters:

KCAR i (i = 1, 2, 3) is the category of units of information containing the selected variables, expressed as the first sequence number of the category (see 4.3).

KCAR i should be adjusted with KREL: if W1 is in level 1, KCAR 1 should correspond to a unit of level 1, otherwise unpredictable results will be obtained.

KVAR i (i = 1, 2, 3) is the final serial number of the variable in the corresponding unit of information.

If each word in a unit of information contains only one variable, the above 2 parameters KCAR and KVAR specify the variable entirely, if not, KCOL i and KMOD i (i = 1, 2, 3) must be specified, in this latter case IFPACK = 1 (see 6.1.1).

6.2.4.1 Tests

The tests from GFILE are complementary to the ones done by RECODE (see 4.4). If a test is not satisfied an error message is printed:

*** "IMPLICATION ERROR" followed by 17 words, or "STRUCTURE ERROR" followed by 20 words. See Figure 5.

1-10 The first 10 words of the unit of information (in case of 1 variable per word, they are the first 10 variables)

11 The serial number of the relation

12 The part I of KREL

13-15 The content of W 1, W 2 and W 3 when defined

16 The serial number of the individual in the processing unit

17 The serial number of the event, within the events of the same category for the individual concerned. ***

If one of these 17 variables has more than 5 digits, it is replaced by asterisks.

If an error occurs, the whole processing unit is rejected, however not before all test operations on this processing unit are terminated. Several error messages can therefore be issued for a single processing unit. All units of information which make up the erroneous processing unit are printed on a rejection file, the Fortran data set reference number of which is IUTERR.

The possibilities of testing are as follows:

IREL = 1 calls for a function with FSN = 10.

The test is very general, it is the implication of two propositions.

Let I 1 be the interval AB, A not included, B included

Let I 2 be the interval DE, D not included, E included

A, B, C, D, E, F, are the parameters of the function

1st proposition P 1 is true if C = 1 and W 1 belongs to I 1
if C = 0 and W 1 does not belong to I 1

2nd proposition P 2 is true if F = 1 and W 2 belongs to I 2
if F = 0 and W 2 does not belong to I 2

If P1 is true P 2 should be true: an error message is printed if P1 is true and P2 is not true.

IREL = 4 calls for one or several test values IRES (NTEST = N) W1 being selected, the implication on W2 has to be verified N times by the IRES value: IRES = W2 x 100 + W1. An error message is printed each time when W2 x 100 + W1 ≠ IRES.

Note: If W1 or W2 are greater than 99 only the 2 last digits of these numbers are considered. Example: to check that W1 = 2 implies W2 = 999 the test value is IRES = 9902.

IREL = 5 calls for one or several test values IRES (NTEST = N). A single error message is printed if one of the N equations W2 x 100 + W1 = IRES is verified: in this case W1 and W2 are incompatible values.

Note: Only the last 4 digits of the value W2 x 100 + W1 are compared with IRES. Example: if W1 = 3 is compatible with 998 the IRES value must be 9803.

IREL = 8 An error message is printed when W1 is not equal to W2.

IREL = 12 An error message is printed when W1 is smaller than W2

6.2.4.2 Functions

A function assigns to the variable W3 a value which is a linear combination of the variables W1 and W2. A, B, C, D, E and F are integer constants.

IREL = 1, several functions are possible according to the value of FSN

FSN = 1: W3 = W1
FSN = 2: W3 = A x W1
FSN = 3: W3 = (W1/B) x A
FSN = 4: W3 = (W1/B) x A + (W2/D) x C
FSN = 5: W3 = (W1/B) x A + (W2/D) x C + E
FSN = 6: W3 = A x W1 + B x W2
FSN = 7: W3 = A
FSN = 8: W3 = W1 if W2 is smaller or equal to B
= A otherwise
FSN = 9: W3 = W1 if W2 is different from B
= A otherwise
FSN = 11: W3 = W1 x 1000/W2 if W2 ≠ 0
W3 = 0 if W2 = 0 (calculation of 0/00)
FSN = 12: W3 is the result of the function selected by FSN = 5, for which one of the values A, B, ... or F is replaced by NSDS (see 6.3) when equated to -999.

6.2.4.3 Conditional transfers

IREL = 1 with FSN = 8 and FSN = 9 (see 6.2.4.2)

IREL = 6 The value W1 is transferred into W3 when W2 = IRES. This operation calls one value of IRES (NTEST = 1)

IREL = 7 The value W1 is transferred into W3 when W2 is greater than IRES. This operation calls one value of IRES (NTEST = 1)

6.2.4.4 Relation table

IREL = 2 The value of W3 is calculated as a function of W1 and W2. The values of W3 are described in a 10 x 10 table. W1 and W2 should therefore be integers in the interval 1-10.

Each table is described by 10 punch cards, as shown on FORM E.

IREL = 11 is used for the recoding of a 2 digits variable. The unit digit is placed in W2, the tens digit in W1, this in a table of rank NWR.
(FORM E) W3 = NEWV (W2, W1, NWR), the rank NWR being identified by the program.
(Note: the 2 digits of the variable have to be increased by 1: for each digit 0-9 should be converted to 1-10, using 2 intermediate variables.)

6.2.4.5 Counting

It is possible to count the units of information of a given category having one or several selected codes in a selected variable. The value 1 is added to the counting variable when the unit of information satisfies one or several conditions, no initialization to zero of the counting variable is performed. An initial value of zero has to be assigned to W3, unless W3 is selected among the new variables, as defined by the extension of the length of the unit of information (see 6.1.4.4). As initialization with zero is not done, consecutive operations allow for the counting of units of information relating to several categories.

IREL = 9 calls one value of IRES (NTEST = 1). W3 = Number of units of information for which W2 equals the selected IRES value.

IREL = 10 calls for NTEST values of IRES, NTEST being an even integer.
W3 = Number of units of information for which W2 is greater or equal to the first IRES value and smaller than the following IRES value, W2 being therefore in one or several intervals.

6.2.4.6 Correspondance between operations, functions, tests and tables

- Each relation is specified on one punch card (FORM C)
- The IRES values (test values) which are called by the relations have to be written in the sequence of the relations, consecutively. For example, if the first relation calls 5 IRES values and the fourth relation 1 IRES value, this latter value will be the sixth in the list of IRES values (FORM D)
- Each table is described on a series of 10 cards (FORM E), this series being in the sequence of the call from the list of relations.
- Similarly, the functions (FORM E2) have to be in the sequence of the calls from the list of relations.

6.3 Organization of the parameter cards

6.3.1 For a single file

- 1 card, FORM A (see 6.1.1, 6.1.4.2, 6.1.4.4, 6.2.4.1) with

  R = 0

  NRUNS = 0

  S = 0

(-) 1 card, FORM AA if NEWFIL = 1 or 2 (see 6.1.2)
- 4 cards, FORM B (see 6.1.3, 6.1.4.2, 6.1.4.3, 6.2.1)
(-) NREL cards, FORM C (see 6.2.4)
(-) 0-25 cards, FORM D, according to the number of IRES values (see 6.2.4.1 and 6.2.4.5), this number being 500 at most
(-) 10 x NNEWV cards, FORM E (see 6.2.4.4)
(-) NNEWF cards, FORM E2 (see 6.2.4.1, 6.2.4.2)

6.3.2 For a multiple file

- 1 card, FORM A (see 6.3.1) with R = Fortran data set reference number for the temporary storage of parameter cards
S = 1 for the first subfile
NRUNS = 1

- other parameter cards (see 6.3.1)
- 1 card, FORM A, for each of the following subfiles with
R = Fortran data set reference number for the temporary storage of parameter cards.
S = 0
NRUNS = 1 except for the last subfile for which NRUNS = 0

In the case of a multiple file NSDS can be a different value for each subfile. This permits, for example, the recording of the date of a subfile, and the using of this date for the calculation, for example, the age of the population for the year of tabulation.

7. THE TABULATION PROGRAM GTAB

The program tabulates information from a Z or a ZZ file, it executes after each reading of the file a classification including 24 000 cells or 2 or 3 classifications of similar nature including 8000 cells. The classification refers to one or several hierarchies. It is in fact possible to classify units of a lower hierarchy as a function of information read in units of a higher hierarchy (standard classification, see 7.3) or inversely.

It is however not possible, in a single classification, to read from various categories of units of information at the third level. This restriction is not valid if a) the categories at third level never contain more than one unit of information, b) the categories of level 3 contain the same number of units, these series of units being related (such as chronological series or coupled events). The categories of units of information with the sequence number following that on level 2 can be defined as an extension of the second level (hierarchy), these categories contain in this case not more than one unit of information. For example one individual can have as extension 2 cards (categories 3 and 4) located in hierarchy 3. The cards 3 and 4 being optional, the extension is of variable length.

It is not possible to classify units of a category of level 3 with reference to the extension of hierarchy 2 (classification of uncoupled events belonging to 2 categories of hierarchy 3). It is however possible to create an extension of fixed length and to consider this latter as a single unit of level 2, either at the time of the definition of the structure of the file Z (GTAB), or at the time of the specification of the parameters of a tabulation variable. If, for example, categories 2, 3 and 4 contain respectively 10, 20 and 30 variables, they can be considered as forming one single unit of level 2 and the variable in position 5 on the unit of the fourth category can be described as the variable of rank 35 in the unit of category 2.

The length of the extension is defined in the first parameter card to GTAB (7.1.1). The standard classification referring to units, for which the sequence number is not greater than the sequence number of the last category of the extension is then performed on the second level. For example, if cards 2, 3 and 4 are individual cards, categories 3 and 4 being defined as extension of category 2, the standard classification will be done according to the name of the

second hierarchy (individual, patient, mother, ...). If categories 3 and 4 are not considered as extensions of category 2 a standard classification will be done according to the name of the third hierarchy (event, birth, pregnancy, ...) (see 7.3).

7.1 Parameter cards for the characteristics of the project and of the input files

Parameter cards are defined below in the sequence in which they are read. The same cards are used for the description of a Z and a ZZ file. For the coding of these cards the file ZZ has to be considered as a Z file for which the structure is 0 card in level 1, 1 card in level 2 and 0 cards in level 3.

7.1.1 Card 1

Columns 1-20 name of hierarchy 1
21-40 name of hierarchy 2
41-60 name of hierarchy 3
61-64 length of the name in columns 1-20 (half word)
65-68 length of the name in columns 21-40
69-72 length of the name in columns 41-60
73-76 number of categories of units of information in the extension of hierarchy 2. Leave blank or code zero if no extension. Code 1 if the extension contains 1 category, i.e. if the number of categories in level 2 and its extension is 2.

7.1.2 Form A (1 card)

MAXTAB : number of tables to be executed

NRUN : not used

NFV : number of independant tabulation variables (defined in FORM D, see 7.2.2)

D : Fortran data set reference number of the input file Z or ZZ (output from GFILE or RECODE)

NSBL : number of subblocks in file D (see also LSBL)

LSBL : number of words in a subblock of file D. If the file is of type Z NSBL x LSBL = 1 x 1820

IP : blank unless file D is packed

IPST : blank unless a packing parameter table is used (ad hoc or output of GPACK)

KCH : number of categories of units of information in hierarchy 1 (0-20)

KTYP : number of categories of units of information in level 3 (0-20)

NWI : number of words in the category of hierarchy 2, excluding the words in the Z zone (introduced in Z) (IWT)

IWT* : number of words in the Z zone (introduced in Z). Code -2 when the file is of type ZZ (exclusion of system words 1 and 2 of a Z file).

MSNFIL : number of subfiles in file D

IWX* : number of words in zone X (excluded from X or ZZ, X being the output of RECODE)

---

* This value should be the same as the one specified in GFILE (see 6.1.4.3)

IERROR : blank, otherwise an abbreviated list of the coding of the stored tabulation variables is printed

LANG : blank or zero: table text in English
1: table text in French

IN** : Fortran data set reference number for the file of tabulation variables:

for a tape file IN $\neq$ 1
for a card file IN = 1

RECFOR : blank or zero: input data file of type Z (output Z of GFILE)
1: input data file of type ZZ (output X of RECODE or output ZZ of GFILE).

7.1.3 Form B (3 cards)

Card B1 : 20 numbers (PNAM) which are characteristic of the 20 possible subfiles. Example: 1970, 1971, etc. If the work is performed in a single file, PNAM (1) appears in the title of each table. Otherwise PNAM (N) appears in the titles of the tables referring to the subfile of rank N.

Card B2 : NWE (20). Number of words in the categories of units of information of level 3, excluding these words in zone X or ZZ. NWE (1) refers to the first category of units of information in level 3.

Card B3 : NWH (20). Number of words in the categories of units of information in the level 1, excluding these words in zone X or ZZ. NWH (1) refers to the first category in level 1.

7.1.4 Other input files

7.1.4.1 Table of t values: 16 cards

7.1.4.2 Packing parameter table: output of GPACK or ad hoc, optional (IPST $\neq$ 0): (n cards)

7.1.4.3 File of tabulation variables: output of VARFIL (tape) or cards (see 7.2)

7.1.4.4 Cards for the specification of tables (see 7.3)

7.2 Parameters for the description of tabulation variables

These parameters are defined in FORM D. Other forms, which are described later (7.2.4 and 7.3.4), are used optionally. The collection of tabulation variables is one of the input files to GTAB. When the number of parameter cards becomes cumbersome, this file can be stored on a tape. The program VARFIL can correct or extend this file, using only the parameter cards concerning the independent variable to be added or corrected.

7.2.1 Card D1, FORM D

Columns 1-28: Name of the variable, which is used by GTAB for the automatic writing of a table title.

Columns 29-32: Length of above name in 1/2 words.

** Do not use reference numbers 3 and 9 which are used respectively for unit 'SYSOUT' and a direct access file for the temporary storage of the tabulation variables. This direct access file is determined by the program statement: DEFINE FILE 9 (1000, 310, U, KK).

Columns 33-36: Serial number of the variable in the file of tabulation variables: start with 1 and increase this number by one unit for each variable. The list of associated variables (FORM E) is considered as a new dependant variable and increases this serial number by one unit (see 7.2.2, ITYP = 4, 5 and 8).

Columns 37-40: First sequence number of the category of units of information to which the variable belongs (4.3).

7.2.2 Card D2, FORM D

ITYP : 2 The variable is used for classification only

3 The variable is used for classification and measurement.

4 Basic variable, has the same structure (or the same categories) as other variables which are listed in FORM E (7.2.4). A basic variable can have at most 60 associated variables, which should belong to the same unit of information. The dependant variable containing the list of addresses for the associated variables must have rank i + 1 in the list of tabulation variables, when the basic variable has rank i. To use the basic and associated variables a third variable of type 6 has to be introduced having a number of categories (codes) equal to the number of associated variables defined in FORM E. Each category or code of this variable of type 6 correspond to the name of an associated variable, the sequence of the names being the one selected on FORM E. The list of names of the associated variables (list variable) can have rank i + 2 in the file of tabulation variables, when the rank of the basic variable is i. This 'list variable' may be used in connection with another basic variable (rank different from i). However the basic variable and its dependant variable cannot be used separately. Not more than one variable of type 4 can be used in one table, and always in connection with a 'list variable'. See also ITYP = 6. This cannot be used jointly with a variable of type 8.

5 Combined variable, has an identical binary structure with at most 6 other variables, and belongs to the same unit of information, the address of the other variables being specified in FORM E. The list of associated variables should have rank i + 1 in the list of tabulation variables if the basic combinatory variable has rank i. The structure of the basic variable is binary and the recoding operations should be used (columns 2-57, FORM D) for associating yes to code 2 and no to code 1. These new codes are not used directly for the classification, but the codes from the associated variables are combined into one single result. This result ranges from 1 to 128, 1 being the code for 'no' to all questions, 128 being the code for 'yes' to 7 questions. A last code (2 xx i + 1, i $\leqslant$ 7) for the classification of these combinations where one or several questions were coded 3 (neither yes nor no). Depending on the number of associated variables the answer 'yes to all questions' will be coded 4 or 8, 16, 32, 128, the answer 'not classified' would be coded 5 or 9, 17, 33, 129. The number of categories which are selected for the tabulation is coded in column 4-5 (LU). The ultimate code (not classified) is included in or excluded from the table, depending on the value given to LU: i or i + 1. The first variable read on FORM E adds 1 or 0 to the combinatory test value, while the last variable read adds at most 64 to this test value. The names of the codes (categories) of this test result are written under CNAM, cards 3 and above, FORM D. For example, if the associated questions to dyspnea are cough and phlegm (presence or absence), the codes or categories of the result can be named ---, --D, -C-, ..., PCD, which mean: none of the 3 signs, dsypnea only, cough only, ..., phlegm and cough and dsypnea. The variables which are listed in FORM E and in the result (CNAM) are the same, but their sequence is reversed. Only one variable of type 5 can be used in one table.

6 List variable (list of the names of the associated variables). It is used with a basic variable of type 4. Its position in a table is the one specified by NVAC (FORM C, columns 57-80, card 2). If such a variable is coded ITYP = 2 instead of 6, the position of this variable in a table is specified by NEWVAR (columns 1-4, FORM E). It contains the names of the associated variables. LU = number of associated variables (columns 4-5). NVACA (columns 66-68) can be the serial number of the basic variable or of any other variable in the same unit of information. The new code which corresponds to the input old code should be greater than zero and less than or equal to LU. In order to meet this condition, the recoding transformations 1-5 can be used. The names of the associated variables are written on card D3 and the ones following D3, in accordance with the indications given under LCOD, NNAM (columns 58-61) and CNAM (D3, ...).

8 Cumulative variable has a common binary or tertiary structure with at most 60 other variables from the same unit of information. The addresses of these associated variables are listed in FORM E. The list variable (see ITYP = 6) must have rank i + 1 in the file of the tabulation variables if the basic cumulative variable has rank i. As the structure of the variable is binary or tertiary, recoding transformations 1-5 can be used in the view of associating yes with code 1, no with code 2, and unknown with code 3. As a result, the cumulation of 'no' on all associated variables is 'no', code 2. The presence of a single 'yes' among all answers is interpreted as 'yes', code 1. A mixture of 'no' and 'unknown' as a final result, code 3. A single variable of type 8 can be used in one table. It cannot be used jointly with a variable of type 4.

MED: This can be used with ITYP = 1 or 3, remains blank otherwise
MED = 0 to 98: MED is added to the measurement
MED = 99: the measurement is divided by ten and rounded to the closest integer.

LU: The highest code shown in the table and accepted for classification. Any new code which is above LU or below 1 eliminates the unit of information to which it belongs from the classification. When LU = 1 the variable is a filter.

Recoding operations

Five transformations can be used. They are performed in the sequence indicated by FORM D: ICCL, MT, NOC, ICC, KCCL

Transformation 1 (translation)

ICCL: The old code ICCL is converted to 1. Inversely, the new code is obtained by adding the increment 1-ICCL to the old code.

Transformation 3 (substitution)

NOC: Number of 1 digit codes to be modified by substitution.

MCOD: Old codes (1 digit), in ascending sequence, consecutive, starting left (column 10).

NCOD: New codes (1 digit) replacing corresponding old codes: NCOD (i) is the new code for MCOD (i).

Transformation 4 (groupings)

ICC: Number of classes of code groups (0 to 3)

ILX: Lowest code of each class

ICR: Intra-class increment

ILL: Lowest code of the last class, the last class being without upper limit.

In order to obtain the age groups 0, 1, 2, 3, 4, 5-9, 10-14, 15-19, 20-29, ..., 70-79, 80+ the coding should be

ICC = 3
ILX = 0, 5, 20
ICR = 1, 5, 10
ILL = 80

The first test: is the old code smaller than ILX (1) + ICR (1)? If yes, the new code is 1, otherwise the new code is more than 1. The following test values are obtained as the addition of the increment to the preceding test value. The above old code groups are replaced by new codes starting at 1, ending at 15.

Transformation 2 (module)

MT: If MT is coded 5 (for example) the old codes 1-5, 6-10, 11-15, etc. are converted to the new codes 1-5, 1-5, 1-5, etc... If this is done in connection with a transformation of type 4 on the first series, producing codes 1, 2, 3 for another tabulation variable on the same file variable, the distribution of a variable including several hundreds of categories can be expanded in several subtables (3 digits codes; occupation for example).

Transformation 5 (translation) (columns 69-70)

KCCL: Two digits constant (or a negative 1 digit constant) can be added to the 'last' old code (intermediate code after transformation 1-4). Example: Add (-1) for the elimination of the first useless age group.

Description of the variables: text and position

LCOD: The name of each category (code) extends over 4, 8, 12 or 16 bytes.

NNAM: Number of names describing the tabulation variable: a name for each category, a name for the total and a name for 0/00. In the table of type 5 (rates) the variable for which rates are calculated should be coded 'yes', 'no', 'unknown', '0/00 yes', '0/00 no', '0/00 unknown'. The positions of yes and no can be reversed. NNAM should be less than or equal to 260 (see 7.2.3).

MWORD: Blank, otherwise the data are packed, and MWORD indicates the word serial number in the unit of information.

MCOLU: Blank, otherwise the data are packed, and MCOLU indicates the position of the variable in the above word.

NVACA: The serial number of the variable in the unit of information. It corresponds to the serial number established by RECODE (new sequence), or to the position of a new variable calculated by GFILE. This number does not depend on the length of the zones. If the data are packed NVACA is the packing module code (see 8.4).

7.2.3 Card D3 and subsequent cards (FORM D)

The number of these cards is determined by the product LCOD x NNAM (columns 58-61, card D2). For example if LCOD x NNAM = 8 x 15 = 120 columns, two cards are necessary. The number of cards is greater or equal to 1 and less than or equal to 13. At most 260 categories are described, each by 1 word (4 characters). The descriptions of the categories

are of fixed length and 4 options are available: 4, 8, 12 or 16 characters for the description of one category. If a length of 8 characters is selected, the texts for each category terminate in columns 8, 16, 24, 32, etc...

7.2.4 FORM E

This is used subsequently to FORM D, when the variable in FORM D is of type 4, 5 or 8.

NEWVAR: If ITYP = 4 and if the variable containing the list of the names of the associated variables is not coded with ITYP = 6, NEWVAR indicates the position of the latter variable in the table. This should however be consistent with the information given by NVAC (FORM C2, columns 57-80).

LISMAX: Total number of associated variables, the basic variable being included.

MWORT: Serial number of the associated variables, one number per 'word'. The sequence of the associated variables is the same as the one described in the list variable. When data are packed, but the packing parameter table is not used and no storage of tabulation variables is performed (NFV = 0), three words are taken for the description of each variable: word, position in the word, module code (using at most 10 cards).

7.3 Parameters for the specification of tables

The standard table is a classification by 1 to 8 variables into at most 24 000 cells. If the variables selected for the execution of one table belong to different hierarchies the unit(s) of the lowest hierarchy are classified. The parameters used for the production of a standard table are described in 7.3.1. Optional parameters for tables referring to a single distribution into at most 24 000 cells are described in 7.3.2. Tables referring to 2 or 3 distributions of similar nature but on different hierarchies are described in 7.3.3.

7.3.1 Standard table: 2 cards per table (FORM C). Card 1 is blank. Card 2 has in columns 55-56 the characters ST (Standard) and in columns 57-80 (NVAC) the serial number of the selected tabulation variables. ST indicates that only cards 1 and 2 are used for the specification of one table.

7.3.2 Table with a single distribution, using non-standard options: 2 or 4 parameter cards per table (FORM C, FORM F). In FORM C the answer 'blank' is converted to a standard code, this code being underlined in the text (see page 74).

7.3.2.1 FORM C, cards 1 and 2

ICALCP 0 frequency distribution

1 frequency distribution and marginal totals

2 code 1 + marginal 0/00

3 code 0-2 and a table for cumulative per thousand distribution and a table containing TOT, SX, SX2, XBAR, VARX, SD (total, sum of X, sum of X xx 2, mean, variance, standard deviation) for each table column

4 Code 0-2 and 2 additional tables giving the 0/00 by rows and columns

5 Rates for trichotomous questions (yes, no, unknown). The trichotomous variable must be the only column variable. The rates 0/00 yes and 0/00 no have as denominator the sum of yes and no answers.

6 Code 3, but the two tables containing the distributions are not printed.

7 The subtable of rank 3n is a rate: it is the collection of quotients, resulting from the division term by term of the elements of the subtable of rank 3n-2 by the elements of the subtable of rank 3n-1. For the production of this table the variable on which the rate is calculated should be the last of the subtitle. The rate will be written in the third category of this variable and is expressed as category 1/category 2 in 0/00. For example the rate still birth/live birth is obtained by the recoding of the variable 'type of birth' in the FORM D. Still birth will be recoded 1, live birth will be recoded 2, the code 3 being dummy or containing the other categories. Code 3 will be named 0/00.

9 Frequency distribution and correlation coefficient with t test.

ISHOPA Calculation of the parameters of the distribution:
0 : none; 1 : SN (sum); 2 : SN/TOT (0/00)

ISHOPL Printing of table parameters
1 : in the right margin
2 : last line
3 : codes 1 and 2
4 : code 3 and SN as subtable
5 : code 1 and SN as subtable

NAXREP This table can be recalculated after the substitution of variables defined in FORM G (see 7.3.4). 0 or blank indicates no recalculation. MAXREP is otherwise a positive integer.

ISMBL With the value 2 for this parameter, for example, it is possible to print side by side two tables with the same structure (weight and height for example). This is used infrequently, and has not been tested in the last revision of the program. 0 or blank or 1 produce a standard table.

MODU Code 1 : the lines with SN = 0 are not printed. The calculated number of subtables per page is not modified by this option. Code 0 : these lines are printed.

NCH Number of variables selected for column headings. The second and the third variables create categories within each category of the preceding variable. Codes 0-3, 1.

NLH Code 1

NSH Codes 0-4: number of variables selected for subtitles.

KCARA 5 or 6 characters per column: codes 5 or 6.

KOLSO The KOLSO first columns are not printed. 0 or blank: no skip of the first column(s).

KOLON KOLON is the last printed column. Code 0 or blank indicates the printing of all the columns.

TITLE This can be written in full, or only the date and serial number can be written. See NTC for the automatic writing of the title. The title indicates the unit to be classified and the name of the selected variables, a name which does not appear in any other place of the printing. The following sequence is recommended:

1) classified unit (child or patient for example)

2) conjunction for the classification 'by' or conjunction for a list 'for'. 'For' is used in connection with a variable of type 4, when the same table is repeated for a list of variables.

3) Name of the variables in the sequence: subtable headings (subtitles): the first is the last to be modified and the last is the first to be modified.
Column headings: the first variable forms major groups.
Line heading.
Example 1. HR10/5.73 Children by cultural group, age, sex. The cultural group appears in the subtitles, age in the column headings, and sex in the line headings, in so far as NCH = 1. If NCH = 2 the cultural group and the age appear as column headings.
Example 2. HR11/5.73 Children for various examinations by age, presence at examination. In this table each subtable has the same general total.

NTC blank: automatic writing of the serial number, the date and the title of the table.

0 : Title not printed

1 : not used

2 : Title printed (as specified by the user)

3-9 : number of words reserved for writing of the date and serial number. The automatic title follows.

NST 0 or blank: printing of subtitles;

1 : printing of the subtitle of the first subtable;

2 : subtitles not printed

NCT 0 or blank: the column headings are printed
1 : the column headings are not printed
2 : the column headings are printed for the first subtable only

MTC 0 or blank : the title is on the same page, its length is A4
1 : the title is on a new page, length A4
2 : the title is on the same page, length A3
3 : the title is on a new page, length A3.

MST blank: the number of subtables per page is calculated by the program.
0 : for certain tables with more than one distribution (see 7.3.3)
1-9 : number of subtables per page, selected as convenient. In the tables showing the variance or columns and lines 0/00 one subtable contains three parts.

CARD 2 (FORM C)

Column 56-57 If the classifications are standard (see 7.3) the FORM F is not used and ST should be coded in columns 56-57.

NVAC Serial number in the file of tabulation variables. The sequence of these serial numbers determines the presentation of the table and differs from the one selected for the writing of the title. The sequence is:
Variables for column headings (1-3)
1 variable for line heading
Variables for subtitles (0-4), the last one being the one 'changing' most rapidly.
Filters (variables with 1 category only) should be the first subtitles, otherwise they can be used as column or line headings.

7.3.2.2 FORM F, cards 3 and 4

NEL Number of distributions (1-3)

INCEL Unit of classification : 1, 2 or 3 for the hierarchy 1, 2 or 3. It is possible to classify the unit of hierarchy 2 (mother for instance) according to the number of units of hierarchy 3 of a selected category and with a specified characteristic (children under 5 years for example) using KREAD = 3.

LIR 1, 2 or 3 if the variables are in the hierarchies 1, 2 or 3. If the variables of the table are in several hierarchies code 4.

NVS Used in 7.3.3

IS1,..., IS7 Not used
The table in FORM F, columns 23(1) to 41(2) includes 8 "columns" for the parameters referring to the tabulation variables and 3 "columns" for the parameters referring to the distributions in the 3 hierarchies. The reading parameters for different variables can be identical and can therefore be written in the same "column". This table should be coded "columnwise": from 23-24(1) to 34(2), from 25-26(1) to 35(2) etc.

"Column 1"

KCAR KCAR(1) First sequence number of the category of units of information containing the first variable(s) of the table.

IVAR IVAR(1) Number of these first variables in "column 1".

KE1, KE2, KE3 Indices defining the reading cycle in hierarchy 3. KE1: initial rank of the unit of information, zero being the code for the last unit. KE2: final rank, zero being the code for the last. KE3: increment between KE1 and KE2; it is a positive integer. The unit of rank KE1 is read first. The next unit is of rank KE1 + KE3 and the following of rank KE1 + 2KE3. The reading cycle is terminated when the rank of the unit to be read is greater than KE2. If the cycle is not used i.e. when hierarchy 3 is not used or if only one unit of information is read code KE1 = KE2 = KE3 = 1. Except for cases where more than 1 unit is read (KREAD = 3) or when a selected unit is read (the first or the last or the nth) the reading cycles in hierarchy 2 and 3 are not used.

I1, I2 Reading cycle on hierarchy 2. I1: initial rank, I2: final rank, zero being the code for the last rank. The increment between I1 and I2 is always 1. If this cycle is not used code I1 = I2 = 1. The simultaneous use of the cycles I and KE gives the possibility of reading all the units of level 3, in the sequence of the data file before proceeding to the classification: this possibility has not been applied.

KREAD

1 Variable used for classification and measurement. The last variable which is coded KREAD = 1 gives the value for the measurement (weight for instance). If none of the variables are coded for measurement, the one which is in the position of the line heading gives the value to the measurement. In correlation tables a supplementary measurement is given by the first variable of the table. In this case IVAR(1) = 1.

2 Variable for classification only

3 To be used only when units of a higher hierarchy are classified according to the number of units of the same category in a lower hierarchy. Only these units are counted for which the selected variable is recoded as 2. Specify IVAR = 1.

4 Variable containing the result of an operation calculated on at most 7 associated binary variables defined in FORM E. See 7.2.2, ITYP = 5, IVAR = 1.

5 Variable which can be associated with 59 other variables with the same structure (FORM E) and used in connection with the 'list variable' containing the names of the associated variables. See 7.2.2, ITYP = 4 IVAR = 1

6 If this variable is on a 'missing' unit of level 3, the value which was read on the last 'present' unit is kept. IVAR = 1.

8 The value of this variable is 1 (yes), 2 (no), 3 (unknown), if the answers in the associated variables (FORM E) are 'at least 1 yes', 'only no', 'other'. IVAR = 1. See 7.2.2, ITYP = 8.

Following "column"

Can be left blank if the reading parameters for the table variables have all been specified in the preceding "column(s)".

"Column 9": classification

KCAR(9): category of information which should be classified. The definitions of the classification cycles are the same as the ones referring to the reading cycles, with the exception of KE1(9) which cannot take the value zero. Unless only the last reading is selected it is not possible to make a classification from the reading in different categories of units in the hierarchy 3. If only the last unit is considered one event only is classified KE1(9) = KE2(9) = KE3(9) = 1 and the reading is performed on the last unit: KE1(i) = KE2(i) = 0 and KE3(i) = 1. When the last unit of information in hierarchy 3 is missing, no classification is done, unless the variable which is read on this unit is of type 6 (KREAD = 6).

7.3.3 Tables with several units of classification: 4 cards per table (FORMS C and F)

ICALCP = 8 Calculation of the mean in each cell of a third 'subtable' (C). The variable, for which the mean is calculated, appears in the three 'subtables' A, B and C as a filter variable (variable with only one category). This filter transforms all the acceptable measurements into the new code 1, the units with unknown or erroneous measurements thus are not classified (new code different from 1).

MST Code 0 if the next 'subtable' is to be presented on the same page.

NEL Code 2 or 3 according to the number of classifications (or distributions).

INCEL(2) = 4 SX (sum of the measurements) in the 'sub-table' B (second classification), simultaneously with ICALCP = 8 (FORM C).

INCEL(3) = 5 A/B: the rate or the mean, is the result of the division of one value of A by the corresponding value of B, in 0/00. It is written in subtable C.

INCEL(3) = 6 B/A: inverse of the preceding result.

NVS In certain tables, one or several filters can be eliminated at the time of the second classification. For example, if NVS = 2, the readings specified by the parameters in 'column' 2 (25-26(1) to 35(2)) take the value 1.

The second classification is therefore independant from the variables specified in 'column' 2. Example: Children under 5 years ('subtable' A), mothers (B), number of children to a mother (C). The denominator (B) contains all mothers, whatever is their number of children, while the numerator (A) contains only children which are less than 5 years. The age of child is selected as filter (subtitle) in the 'subtable' A. This filter is eliminated when 'subtable' B is calculated.

7.3.4 FORM G

The use of this form has not been tested in this revised version of the program. This form is used when a next table can be obtained by the shifting of one or several variables (results for consecutive examinations for example). In this case MAXREP indicates the maximal number of tables to be produced: for one repetition code MAXREP = 2 (FORM C). One or several forms G have to be read immediately after the 2 or 4 parameter cards for this table (FORMS C, (F)). The number of forms of type G is equal to MAXREP - 1. When the translation refers to only one variable the use of the special variables 'basic variable' and 'list variable' is recommended (see 7.2.2).

MVSN, MMOT, MCOLO, MPILI (columns 1-7) :

MVSN: Serial number of the variable in the table (code 1-8)

Unpacked data: MMOT serial number of the new variable, MCOLO and MPILI are blank

Packed data: MMOT, MCOLO, MPILI: word, position in the word, module for the new variable.

Columns 50-77 new title for the table

NTC = 0: no title
= 1: title is terminated on this card
= 2: title is continued on the next card

MTC = 0: the table title is on the same page
= 1: the table title is on a new page

If a second card is used: columns 1-56 contain the continuation of the title.

8. OPTIONAL PROGRAMS

The purpose of these programs has been described briefly at the end of chapter 1. The detailed description and the use of these programs is given in the following paragraphs.

As a general rule, the parameter cards for these programs contain at most 20 integer parameters, right adjusted in consecutive fields of 4 columns. Exceptions to this rule are noted in the detailed descriptions.

8.1 Edited printing of the parameter cards

The FORM A is a manuscript document on the content of a project. It can be stored on cards or tape and constitutes the file of the variable names and the variable sequence numbers. This file is used as input to RECODE, with MODCOR2 = 2 and to LISCOD. In the above two utilizations, the file does not contain these parameter cards which define the way in which unused fields are to be skipped or the return to a field for a new reading (two or more new variables for one old variable).

8.1.1 LISCOD

This program presents in print form the parameter cards for RECODE and the parameter cards forming the file of the variable names. It prints, for each category of units of information a table showing the correspondance between the old and the new codes, and the intermediate and new sequence numbers for each variable.

Input parameter cards:

Card 0: Project title (80 characters)

Card A: 1 card of format 20 I4 containing

NV1 number of variables of 1 character
NV2 number of variables of 2 characters
NV3 number of variables of 3 or more characters
NVS number of skips
NEWCO: code 0: sequence of the variables in file ZZ is unchanged
code 1: sequence of the variables in file ZZ is modified
NRUNS: code 0: only one category of units of information or the last category
code 1: the category of units of information KARTYP is followed by another category
KARTYP: first sequence number of the category of units of information
NEC: number of words in a subblock of the output file record from RECODE (of length NBL x NEC $\leqslant$ 1820). This is equal to the number of variables contained in the longest unit of information, including the new variables (see 5.2).

Cards B: Name and serial numbers (intermediate and final) of each variable, including new variables, excluding 'skips'. (FORM A)

Cards C: Parameter cards for recoding (FORMS C, D, E, F, (G)), the cards corresponding to FORM G being inserted only when NEWCO $\neq$ 0.

The above defined parameter cards A, B, and C refer either to the first category of units of information (KARTYP = 1) or the following categories. The last category of units of information is specified by NRUN = 0, the preceding ones by NRUN = 1 when they exist.

The sequence of the parameter cards is:

Card 0
Cards A1, B1, C1
Cards A2, B2, C2
.....
Cards An, Bn, Cn

n is the number of categories of units of information.

8.1.2 LISVAR

This presents in edited print form the parameter cards to GTAB referring to the tabulation variables. It uses the cards corresponding to FORM G of RECODE, when the sequence of the file variables has been modified (NEWCO $\neq$ 0). It is made from two programs: the first program presents the parameter cards in the storage sequence; the second program presents the same variables but now according to the questionnaire sequence. The two programs are used jointly in the same job including 3 steps: LISVAR1, SORT, LISVAR2, the sorted output of LISVAR1 being the input to LISVAR2.

8.1.2.1 LISVAR 1

Input parameter cards:

Card 1 NFV: Number of independant variables in the file of tabulation variables.
IERROR: Code 1: printing of an abbreviated list of the parameter cards (forms D, (E) of GTAB)
Code 0: no printing
IPST: Not used
IP: Code 0: unpacked variables
Code 1: packed variables
IN: Fortran data set reference number for the file of tabulation variables
Code 1 if this file is on cards.
Format of card 1 is 20 I4.

Card 2 Title of the printed document. Suggested content: 'Name of project, list of variables, storage sequence, date'.
Format 20 A4.

Cards 3-N or tape: File of the tabulation variables (Forms D, (E) of GTAB):
a) cards, with IN = 1 on card 1
or b) tape, output of VARFIL (see 8.5)

Cards A-B: Option for new sequence: if for certain categories of units of information the sequence of the variables on the file ZZ is not the original sequence (NEWCO $\neq$ 0 in RECODE) additional parameter cards A and B have to be added to each of those categories:

Card A NV1: number of variables of 1 character
NV2: number of variables of 2 characters
NV3: number of variables of 3 or more characters
NS: number of skips
NEC: total number of variables including the new variables (see 5.2 and 5.7.1 RECODE)
NEWSEQ: Code 1: new sequence of the variables
NRUN: Code 0: last category of units of information
Code 1: another category will follow
NSPEC: first sequence number of the category.
This card A is of format 20 I4.

Card B Cards for the new sequence of the variables:
FORM G of RECODE.

The above described cards (A and B) refer either to the first category of units of information (NSPEC = 1), or to one or several other categories, the last category being specified by NRUN = 0, and the preceding ones by NRUN = 1 when they exist.

The sequence of the input parameter records is:

Cards 1 and 2
Cards 3-N or magnetic tape
Cards (A1, B1))
Cards (A2, B2)) optional
.....)
Cards (Ai, Bi))

with i = the number of categories of units of information for which NEWCO = 0.

8.1.2.2 SORT

LISVAR1 creates two intermediate files: the file with the Fortran data set reference number equal to 9 (direct access) including 1000 records of 1260 bytes for at the most a thousand variables and the sequential file with the Fortran data set reference number equal to 12 and DCB = (RECFM = VBS, LRECL = 1256, BLKSIZE = 1260). The last file 12 is the input to the SORT program. The output sequence is given by:

a) the first sequence number of the category of units of information (variable 10 in file 12 (called CARD)

b) the original serial number of the variable in the file ZZ (Variable 8 (OSER))

c) the serial number of the tabulation variable in the storage sequence (variable 11 (VAR))

The parameter card to the SORT program is therefore:
bSORT FIELDS = (41, 4, A, 33, 4, A, 45, 4, A), FORMAT = BI.

8.1.2.3 LISVAR 2

Only one input parameter card:

Title of the document with the suggested content:
'Name of project, list of variables, questionnaire sequence, date'.

8.1.3 LISTAB

It presents the parameter cards of GTAB which refer to the specification of the tables. Standard options and the table title are edited by this program. The calculation of the parameter which specifies the number of subtables per page is executed (MST, form C of GTAB). It edits:

a) a list of parameter cards for the specification of the tables in the sequence of the tabulation plan

b) a table describing the title of the tables referring to each tabulation variable, the sequences being the one of the tabulation plan and the one of the tabulation variables in storage.

Input:

Card 1: format 20 I4

IN = Fortran data set reference number for the file of tabulation variables. Code 1 for cards. IN $\neq$ 9.
INT = Fortran data set reference number for a temporary disk sequential file (LRECL = 136, RECFM = VBS). INT $\neq$ 9.
NFV = Number of independant tabulation variables
IERROR = Code 1: abbreviated print of tabulation variables
Code 0: no print
IP = Code 0: unpacked variables
Code 1: packed variables
NVOP = Code $>$ 0: print table b only, starting with the tabulation variable with the storage serial number equal to NVOP.

Card 2:

columns 1-20 Name of the unit in hierarchy 1
columns 21-40 Name of the unit in hierarchy 2
columns 41-60 Name of the unit in hierarchy 3
columns 61-64 Length of name 1 (in 1/2 words)
columns 65-68 Length of name 2 (in 1/2 words)
columns 69-72 Length of name 3 (in 1/2 words)

The format of this card is therefore 3(5A4), 3I4; information should be left adjusted for the alphabetical part and right adjusted for the numerical part.

Card 3: Title 1 with the suggested content: 'Project name, tables , date'.

Card 4: Title 2 with the suggested content: 'Project name, tables by variables , date'.

Tape or cards: tabulation variables

Cards: Specification of parameters for the tables, in the sequence of the tabulation plan. For each table either 2 cards (FORM C, GTAB) with ST in columns 55-56 of card C2 or 4 cards i.e. 2 cards (FORM C), card C2 is not coded ST in column 55-56, and 2 cards corresponding to FORM F.

Note: A temporary direct access file, with the Fortran data set reference number equal to 9 is used during the execution of this program. It is defined by the statement DEFINE FILE 9(1000, 310, U, KK). The number of tabulation variables should therefore not exceed 1000.

8.2 GREV

This can modify the value of the variables of sequentially identified records (units of information) in a ZZ file, identification being specified by at most ten selected words or identifiers, which may be any word of the record. It is possible also to modify the identifiers. The error lists from RECODE, after their submission to corrections (in old or new codes) and punching, constitute one input to REVISE (the correction file). The identifiers used for REVISE are in this case the ones used by RECODE. When old codes are used for the corrections, these old codes are checked and transformed to new codes according to the parameters of RECODE (FORMS C, D, E, F, (G)). REVISE produces a revised ZZ file and as an option a list of records containing one or several corrections. The corrections which are refused by the tests of RECODE are indicated by stars (****). The sequences in both the basic ZZ file and the correction file must be in harmony. A test (with option ITRI = 1) is done on the identifiers of the correction file (IC) and on the identifiers of the basic ZZ file (IB). If during revision IC is smaller than IB the process is interrupted. The correction card which has no corresponding record on ZZ file is printed with the message 'missing record on ZZ file' and the next correction card is read and its identifiers are compared with the identifiers of the actual record in the ZZ file. If the tests on the relative values of identifiers are not performed (ITRI = 0) but only the equality of identifiers is checked, the process will continue even when IC is smaller than IB. In this latter case the IC = IB might never be met and the process stops with the error message 'erroneous identification'. The basic ZZ file might be divided in subfiles, each subfile can contain several categories of units of information. If a ZZ basic file contains records with identical identifiers this file ZZ can be submitted to consecutive subrevisions, which give the possibility of correcting the identifiers without preliminary sorting operation (8.2.4).

Two types of revisions can be performed:

a) the file ZZ is sorted with the category of units as the primary sorting variable (sequence of the ZZ file as output of RECODE); in this case the file ZZ is read only once;

b) the basic file ZZ is sorted according to the identifier of the processing unit (household number for example) (ZZ as input or output of GFILE). It is read and partially revised as many times as the number of categories. The partially revised file of rank i is the input to the partial revision of rank i + 1, each partial revision concerning only one category of units and the totality of the file.

8.2.1 Input parameter cards to REVISE:

The parameter cards of type A, B, C, D, and E are described below:

Card A Columns 1-68 Title
Columns 69-72 NEW = 0: old code
NEW = MAX (NVAID): the maximal number of identifiers to be tested (FORMS B2, RECODE, col. 1-4): new code
Columns 73-76 INT: Fortran data set reference number for intermediate storage.
INT = 0 or blank: no storage used and NFDIR = 0
INT = 13: storage used and NFDIR is not zero
Columns 77-80 NFDIR: number of categories of units to be stored.
Format of card A (17A4, 3I4).

Card B 1) INC: Fortran data set reference number of the correction file ($\neq$ 13)
2) IN: Fortran data set reference number of the basic ZZ file
3) NBLIN: number of subblocks in the ZZ record (IN or IUT)
4) LBLIN: number of words per subblock
5) IDIM: code 0: the logical record length is IDIM = NBLIN x LBLIN
code 1: IDIM = 1820
6) ITRI = 0 Unconditional search of IB in the basic ZZ file
= 1 Search of IB insofar as IC is smaller or equal to IB (IB and IC see 7.2)
7-10) NV1, NV2, NV3, NVS
11-15) NEND, KEND, IUT, NRUN, NPRINT
16-20) NEWCO, NSPEC, NSFIL, KTYPO, NRSTA
See explanations in FORM B (RECODE) or below:
14) NRUN: 0 (1 category of units of information and simple file) or last category
1 several categories or multiple file, except for the last processing
2 Rewind IN and IUT with NRSTA = 0
15) NPRINT: 0 no printing
1 printing of corrected records
18) NDFIL: 0 no end of subfile on the output ZZ file (IUT)
1 end of file at the end of the subfile on ZZ
19) KTYPO: Intermediate serial number of the file variable 'sequence number of the category of units of information' (see 5.3.2)
20) NRSTA: code 0: processing of the new category when the file or subfile is terminated
code 1: processing of the new category when the category is modified. The end of category i is detected by the reading of the first card of category i + 1. A backspace allows for the revision of the first card of each following category.

Card C 1 card for the selection of the identifiers:
FORM B2 (RECODE), first card only.

Card D Parameter cards for RECODE, forms C, D, E, F; only if NEW = 0 or NEWCO $\neq$ 0.

Card E Parameter cards for RECODE: form G only if NEWCO is not zero.

Card F One record for each corrected variable. The parameters are: Serial number of the correction: it is not read by REVISE but printed by RECODE; it can be used as a reference to the error lists established by RECODE.

ITEM1: Intermediate serial number of the variable to be corrected.
NSCA: code 0 or blank: new record
code 1: same record
IDR: intermediate serial numbers of the variables used as identifiers. The number of identifiers is taken as 10 when the corrections are expressed with old codes, as NEW with the new codes (see above 8.2.1, Card A).
AA: Correction in old code: right adjusted in the field 73-78.
NX: Correction in new code: right adjusted in the field of 6 characters which follows the field for the identifier of ultimate rank NEW.

The formats for the reading of the correction records are in old codes: (6X, I5, I1, 10I6, 6A1), in new codes: (6X, I5, I1, 11I6) see also 5.5.2.

8.2.2 Sequence in the basic file ZZ and in the correction file

If ITRI = 0 the correction records have to be in the same sequence as the erroneous records on the basic file. A sort according to the category of units of information of the file of corrections is however necessary. If ITRI = 1 both the correction and basic files should be sorted according to the identifiers.

In case of multiple files the category of unit of information belongs to the minor sorting variables.

8.2.3 Sequence of the parameter cards

The following cases are demonstrated:

A single category of units of information, simple file:

```
Cards                A B C (D) (E) F

Specificities for  A   INT = 0, NFDIR = 0
              for  B   NSPEC  = 0, NRUN = 0
                       NRSTA = 0
```

A single category of units of information, multiple file:

```
Cards                A B1 C1 (D) (E1) F1
                       B2 C2     (E2) F2
                     ...................
                       Bm Cm     (Em) Fm
```

m being the number of subfiles

```
Specificities for  A  INT = 0  NFDIR = 0,
              for  B  NSPEC = 1 for B1
                           = 0 for B2, ..., Bm
                      NRUN = 1 for B1, ..., Bm-1
                           = 0 for Bm
                      NDFIL = 1, NRSTA = 1
              for F:  insert a blank card at the end of each Fi
```

Several categories of units of information, simple file:

```
Cards                A B1 C1 (D1) (E1) F1
                       B2 C2 (D2) (E2) F2
                     ....................
                       Bn Cn (Dn) (En) Fn
```

n being the number of categories

```
Specificities for  A  INT = NFDIR = 0
                   B  NSPEC = 1
                      NRSTA = ITRI = 1                  )
                      NRUN = 1 for B1, ..., Bn-1        )  type a (8.2)
                           = 0 for Bn                   )
                      NRSTA = ITRI = 0                  )  type b (8.2)
                      NRUN = 2 for B1, ..., Bn-1        )  (see also 8.2.4)
                           = 0 for Bn                   )

                   F: insert a blank card after each Fi
```

Several categories of units of information, multiple file:

```
Cards                A BO, 1 D1     )
                       BO, 2 D2     )
                     ..........     )      Disk storage
                       BO, n Dn     )
```

```
                ( B1,1 C1,1 (E1,1) F1,1
     Subfile 1  ( ......................
                ( B1,n C1,n (E1,n) F1,n

                ( Bm,1 Cm,1 (Em,1) Fm,1
     Subfile m  ( ......................
                ( Bm,n Cm,n (Em,n) Fm,n
```

m being the number of subfiles, n the number of categories of units.

```
Specificities for A:  NFDIR = n (number of categories)
                      INT = 13
                   B: NSPEC = first sequence number of the category of units
                      NRUN = 1 for all B except Bm,n
                           = 0 for Bm, n
                      NDFIL = 0 for Bi, j (i = 1,m;  j = 1, (n-1)
                            = 1 for Bi, j (i = 1,m;  j = n)
                      NRSTA = 1
                      ITRI = 1

                   F: insert a blank card at the end of each Fi,j for i = 1,m
```

If a <u>run of type b</u> is wanted the REWIND operation should not be performed before all subfiles have been partially revised, i.e. before the records belonging to one category have been revised in all subfiles. The sequence of parameter cards should be:

```
                ( B1,1 C1,1 (E1,1) F1,1
     Category 1 ( ......................
                ( Bm,1 Cm,1 (Em,1) Fm,1

                ( B1,n C1,n (E1,n) F1,n
     Category 2 ( ......................
                ( Bm,n Cm,n (Em,n) Fm,n
```

m being the number of subfiles, n the number of categories.

```
Specificities for A:  NFDIR = 0
                      INT = 0
                   B: NSPEC = n for Bi, j  (i = 1;  j = 1,n)
                            = 0 for Bi, j  (i = 2,m;  j = 1,m)
                      NRUN = 1 for Bi, j   (i = 1, m-1;  j = 1,n)
                           = 2 for Bi, j   (i = m;  j = 1, n-1)
                           = 0 for Bm, n
                      NRSTA = 0
                      ITRI = 0
                      NDFIL = 1
                      IN = 11 for Bi, j  (i = 1,m;  j = 1
                           12 for Bi, j  (i = 1,m;  j = 2K)
                           14 for Bi, j  (i = 1,m;  j = 2K + 1)
                      IUT = 12 for Bi, j  (i = 1,m;  j = 2K-1)
                            14 for Bi, j  (i = 1,m;  j = 2K)

                   F: insert a blank card after each of the n Fm,i (i = 1,n) k = 1, (n-1)/2;
```

11, 12 and 14 are examples for Fortran data set reference numbers for input and output units (see also 8.2.4).

8.2.4 Processing of records with identical identifications

A ZZ file in hierarchial sequence, such as the output file of rejected household information from GFILE, can be revised without sorting. The records with identical identifications (if any) keep therefore a known position in the ZZ file. For example, in order to modify the identification of the second record with identification A, two correction records are prepared: the first has the function of passing the first record with identification A on the output file ZZ, the second should modify the identification of the second record with identification A and pass it on the revised ZZ output file. When the basic file ZZ contains several categories of units of information (n), the revision has to be subdivided into n partial revisions, each of them being relative to a single category, the basic file ZZ being therefore processed entirely n times.

The sequence of the parameter cards (see 8.2.3) and the specificities for the cards B are shown by the following example for a simple file containing 5 categories of units:

A	B1	C1	(D1)	(E1)	F1	IN = 11	IUT = 12	NRUN = 2	NRSTA = ITRI = 0
	B2	C2	(D2)	(E2)	F2	IN = 12	IUT = 14	NRUN = 2	NRSTA = ITRI = 0
	B3	C3	(D3)	(E3)	F3	IN = 14	IUT = 12	NRUN = 2	NRSTA = ITRI = 0
	B4	C4	(D4)	(E4)	F4	IN = 12	IUT = 14	NRUN = 2	NRSTA = ITRI = 0
	B5	C5	(D5)	(E5)	F5	IN = 14	IUT = 12	NRUN = 0	NRSTA = ITRI = 0

8.3 DELRET

The program selects those records with the characteristics which are defined in parameter cards. The input parameter cards are:

Card 1 Col. 1-56 Title
Col. 57-80, in 6I4 format:

IN: Fortran data set reference number of the file which is to be analysed. The output file will be written with unit IUT = IN + 1 unless IN is larger than 100. In this case IN = MOD (IN, 100) and IUT = 6.

NBL: Number of subblocks in the input record of the file IN.

LBL: Number of words in one subblock.

MLRECL: Record length for the files IN and IUT
Code 1 MLRECL = 1820
Code 2 MLRECL = NBL x LBL

KREDE: Code 1 selection (inclusion in IUT)
Code 2 deletion (exclusion from IUT)

MSN: Codes 0, blank or 1 simple file
2 or more number of subfiles

Condition cards

The program accepts at most 100 condition cards, each of these cards corresponding to one condition. The parameters are written in format 20I4.

NSER: Serial number of the variable
A: Lower limit of the interval
B: Upper limit of the interval
O: Code 1 X.LT.A
Code 2 X.GE.A.AND.X.LE.B
Code 3 X.GT.A

NCOND: The parameter card belongs to a group of NCOND consecutive conditions which are related by 'AND', the groups of conditions being related by 'OR'.

If KREDE = 1 (card 1) the output file ZZ (IUT) includes the records which satisfy the conditions which are defined in the parameter cards and no other records.

If KREDE = 2 the output ZZ file contains all the records of the input file with the exception of these records which satisfy the conditions which are defined in the parameter cards.

8.4 GPACK

The program transforms a ZZ file (output from RECODE, without option on new sequence, i.e. with NEWCO = 0) into a packed ZZ file. It produces also a table of parameters of the transformation, these parameters being calculated by the analysis of the parameter cards from RECODE (Forms C, D, E and F). The parameter table can be printed on paper or punched on cards: it is used as optional input to GFILE and GTAB. The application has been done however only with simple files, including one category of units of information, excluding the use of GFILE. Some testing and eventual adaptations must be done for applications of more general charcter. GPACK is useful when the ZZ file includes a large number of variables having a small number of categories (di- and tri-chotomous questions for example). The variables of the ZZ file should be positive, different from zero when their base is 2, 3 or 10, zero being acceptable when the base is larger than 10 (see INTYP, card 1).

Card 1 Format 20I4

IN: Fortran data set reference number of the input basic file ZZ
NSBLX: Number of subblocks per record of the input file
LSBLX: Number of words per subblock
IUT: Fortran data set reference number of the output file
INTYP: Code 1: the programmer selects the base for each variable which can be 2, 3, 10, 100, 1000, 10000; for example the base 2 is appropriate for a dichotomous question, the base 100 for a 2 digit measurement. Write and punch in 24 I3 format:
Word serial number (MOT)
Position in the word (KO): see below, code 0
Base (B): see below, code 0 for each variable, according to the sequence of the file variables in the file.
Code 0: the packing parameter table is prepared by the program.
Base B is coded 1, 2, ., 6 for words containing variables with base 2, 3, 10, 100, 1000 and 10000, 7 for words containing no more than 1 variable.
The maximal numbers of packed variables per word for bases 1, 2, 3, 4, 5, 6 are respectively 31, 19, 9, 4, 3, 2.
The positions in the word range from 1-31 for variables with B = 1, from 1-19 for variables with B = 2 etc...
NV1: Number of variables of 1 character
NV2: Number of variables of 2 characters
NV3: Number of variables of 3 characters or more
IPUNCH: Code 0 the packing parameter table is not punched
Code 1 the packing parameter table is punched
NTSTAP: Code 0 test tape not produced
Code 1: Attention: writing on the tape IN.
For testing the use the basic file ZZ should be avoided. The test file consists of three records: record 1 with minimal codes (1 for variables of bases 2, 3 and 10, 0 for variables of other bases); record 2 with maximal codes and for record 3 only the first code is defined and is equal to zero
KBLKY: LSBLY and NSBLY of the output file being determined by the program, KBLKY should be coded 1820 for the first test. This value can be replaced by LSBLY x NSBLY

at the second test, if desirable.

Cards 2-N: Optional, only if INTYP $\neq$ 0

Cards N + 1-M: Parameter cards to RECODE, Forms C, D, E and F

8.5 VARFIL

When the tabulation variables are numerous, it is convenient to store them on magnetic tape, this tape being used as input of parameters to GTAB. The program has to be submitted to compilation in the view of two different utilizations.

1) Creation of the file of tabulation variables:

The statement IN = 11 precedes the statement IN = 1: the parameter cards for the variables are read and stored thereafter.

Input: Forms D and (E) of GTAB in storage sequence. The output is done on tape, with unit IUT = 11.

2) Revision of this file:

Corrections or additions of variables: IN = 1 precedes IN = 11: the tape file is read before the corrections and addition cards.

Input: a) tape file, b) revision cards: form D (complete) and (E). Associated variables cannot be read separately. Output: this is performed on the same tape and the same file, with IUT = 11.

Remarks

The program uses an intermediate direct access file with the Fortran data set reference number equal to nine.

It prints at the end of the execution the number of dependant and independant variables which are stored on the tape.

8.6 ZZFILE

This prints on paper a file of type ZZ.

The input cards are:

Card 1 (format 14A, 6I4)

Columns 1-56: Title
57-60: IN: Fortran data set reference number for the ZZ file
61-64: NMAX: Code 0: Complete listing
Code > 0: the NMAX first records are printed
65-68: NBL: Number of subblocks in ZZ
69-72: LBL: Number of words per subblock
73-76: MLRECL: Code 0 record length = 1820
Code 1 record length = NBLXLBL
77-80: MSNMAX: Number of subfiles
Code 0, blank or 1 simple files
Code 2 or more: 2 or more subfiles

Card 2 (Format of the printout)

Columns 1-80: Fortran Format for the printing on paper of the output record of length LBL

Example: LBL = 20. Write (1X,20I6) in columns 1-9 of card 2.

8.7 ZTOZZ

This converts a Z file into a ZZ file. On option the variables 1 to 3 contain the category of units of information (first sequence number), a serial number for the processing unit (example: the household serial number), an identifier for the units of the second hierarchy (example: the individual serial number). These variables constitute the zone IWA in the output file ZZ which is of 0 to 3 words. A second zone IWX which is of 0-10 words can follow the zone IWA. IWX contains in the sequence which is to be specified by parameters from 0 to 10 words which must be selected in the zone Z (IWT) of the input file Z. The conversion can be stopped after a given number of processing units (MAXHH).

The input parameter cards are:

Card 1 (with format 20I4)

Columns 1-4: IN Fortran data set reference number for file Z
5-8: LSBL Length of the subblock in file IN
9-12: NSBL Number of subblocks in one record of file IN
13-16: KCH Number of categories of units of information in hierarchy 1
17-20: KTYP Number of categories of units of information in hierarchy 3
21-24: NWI Number of words in the unit of information of hierarchy 2, excluding those reserved for identification (zone X of length IWX in the input file X or ZZ to GFILE)
25-28: IWT Number of words in the zone Z in GFILE (IWT)
29-32: MSNFIL Number of subfiles in file IN
33-36: IWX Number of words in the zone of the output file
37-40: IWA Zone for new identifiers in the output file
Code 0 no zone IWA is used
Code 1 First sequence number for the category of units of information
Code 2 Serial number for the processing units (for example household number) and code 1
Code 3 Serial number for the units of hierarchy 2 within the processing unit (individual serial number for example) + codes 1 and 2
41-44: IUT Data set reference number for output file ZZ
45-48: LSBLZZ Number of words in the record of output file ZZ
49-53: MAXHH Code 0 or blank Complete conversion of Z into ZZ
Code N N processing units of file Z are converted into the corresponding file ZZ

Card 2 (with format 20I4)

NWE(20) Number of words in the units of information of hierarchy 3 (excluding the words in the identification zone X + Z) for each of the KTYP categories of units (see GFILE, form B)

Card 3 (with format 20I4)

KWH(20) Number of words in the units of information in hierarchy 1 (excluding the words in the identification zone X + Z) for each of the KCH categories of units (see GFILE, form B)

Card 4 (with format 20I4)

KPO(10) KPO (I): new position within the zone ZZ of the output file ZZ of the word of rank I in the zone Z of the input file (if IWT ≠ 0) (see GFILE, form B)
Code 0 or blank if IWT = 0

8.8 ZMAP

This prints on paper the description of the structure of each of the n first processing units of a Z file. As option the complete information in the n first processing units can be printed, with a fixed format of 20 words per line.

The input parameter cards are

Card 1 (with format 20A4)

Title

Card 2 (with format 20I4)

Columns 1-4:	NPRINT:	code 0	structure only is printed
		code 1	complete information is printed
5-8:	NHMAX:	code 0	the full file is processed
		code n	only n processing units are treated
9-12:	IWT:	number of words in the Z zone as defined in GFILE and GTAB	

Cards 3-6 (with format 20I4)

Forms A, B1, B2 and B3 of GFILE

The printed output contains also some details on the content of the fixed and variable length parts of the Z zone of the Z file.

9. GLOSSARY AND SYMBOLS

Underlined words are explained in the glossary.

1. By: see distribution (see also TITLE, 7.3.2.1).

2. Category of units of information: collection of units of information which have the same format and the same variables. Example: all individual records referring to physical examination (see 4.3).

3. Category of a variable: see variable.

4. Cell: Example: A table resulting from a distribution by a variable of 2 categories and a variable of 9 categories contains 2 x 9 = 18 cells. If we include the subtotal and the total the number of cells will be increased to 3 x 10 = 30 cells (see 7).

5. Classification: see distribution (see 7).

6. Code: a numerical integer value which is in one to one correspondance with a category of a variable.

7. Distribution: when units of information are distributed according to or by a variable or a classification or subdivision of the collection of units of information into the various categories of the variable is performed. Example: a distribution of 100 children by sex into 45 males and 55 females.

8. File variable: a variable which is described by the transformation parameters of RECODE (and GFILE). It might be identical to the questionnaire variable. To one questionnaire variable corresponds generally one or several file variables. To one file variable corresponds at most 1 questionnaire variable unless the new file variable is a combination of several questionnaire variables.

9. Filter: see variable (see also 7.2.2 and 7.3.2.1).

10. For: when units of information are distributed for a group of variables several classifications into the various categories of the variables are performed. Example: a distribution of 100 children for physical examinations result in one distribution by weight and one distribution by height (see TITLE, 7.3.2.1).

11. Hierarchy: when two units of information have a relation of belonging they belong to two different hierarchies. All units belonging to the same category will be on the same hierarchy or level (see 4.2, also 7 for extension of hierarchy). When two categories are on the same hierarchy no relation of belonging between them is specified.

12. Level: see hierarchy.

13. Packed data: data are packed when at least one Fortran word contains more than one file variable. Packing should not be used unless a great number of variables can be included in a few Fortran words.

14. Processing unit: collection of units of information which belong to an identified entity. Example: household record and individual records belonging to this household (see 3.2, 4.2).

15. Sequence number of the category: see 4.3.

16. Tabulation variable: the description of a variable and its categories which is used by GTAB for the production of a distribution. To one questionnaire variable correspond zero or several tabulation variables (see 7.2 and 8.1.2).

17. Unit of information: collection of sequential file variables which belong to an identified entity. Example: individual card or record (see 4.2).

18. Variable: can be either continuous and refers to a measurement (such as weight) or discrete and refers to categories (like yes, no, unknown). If a variable contains a single category it is named a filter.

19. Variable serial number: three levels are considered when it refers to the file variable:

    a) Input to RECODE: initial serial number

    b) Output from RECODE with NEWCO = 0: intermediate (OSER) (OLD)

    c) Output from RECODE with NEWCO $\neq$ 0: final (NSER) (NEW) (see RECODE FORM A and 5.3.2).

20. Zone: see 6.1.4.3.

21. * or x multiplication sign.

22. ** or xx power sign.

23. MOD (IN, 100) (example): the rest of the division of IN by 100.

24. i = 1,m (Fortran) is used as an equivalent notation to i = 1, ..., m (mathematical).

10. FORMS FOR THE CODING OF THE PARAMETER CARDS

General: A;
RECODE: B, B2, C, D, E, F, G;
GFILE: A, AA, B, C, D, E, E2;
GTAB: A, B, C, D, E, F, G.

Revised 16.12.75

RECODE GFILE, GTAB FORM A	DATE STUDY CARD, RECORD TYPE

INPUT TO RECODE				OUTPUT FROM RECODE INPUT TO GFILE OR GTAB				
VARIABLE OR INTERVAL: ITEM				SERIAL NUMBER OF VARIABLE				
CARD COL. OR RECORD BYTE *	NAME	FIELD LENGTH IN BYTES **	SERIAL NO.	OLD ***	NEW ***	NOT TO BE PUNCHED		
COLUMN 1234567	1 2 8901234567890123456 78	3 90	123	456	789			

* THE CONTENTS OF EACH FIELD SHOULD BE DEFINED AT LEAST ONCE
** NOT USED BY THE SYSTEM: FOR A NEGATIVE SPACE THE MINUS SIGN IS INDICATED IN THE TEXT PART
*** CAN BE LEFT BLANK IF NOT DIFFERENT FROM PREVIOUS COLUMN

Revised 12.7.79

RECODE	DATE:
1 + N*2 cards	STUDY:
FORM B	RUN

Card 1: JOB characteristics Title		Card 1 Col
		1-28
		29-56
		57-68
MØDCØR1	Printing of instructions for revision: 1 = NO, 2 = YES	69-72
INT	Fortran serial number for intermediate storage (YES = 13, NO = blank)	73-76
NFDIR	Number of data sets on intermediate storage (1 data set per card type usually)	77-80

Run cards (2 cards per run): Card 2.1		Next Card Col
IN	Fortran serial number referring to input data ($\neq$ 13)	1-4
NBLIN	Number of blocks in the input record	5-8
LBLIN	Number of bytes in one block of the input record	9-12
NCOL	Number of characters (columns) covered by input data	13-16
NV1	Number of variables of 1 character	17-20
NV2	Number of variables of 2 characters	21-24
NV3	Number of variables of 3 or more characters	25-28
NS	Number of gaps	29-32
NEND	If the new code of the KENDth variable in the input file = NEND, the file is terminated	33-36
KEND	The KENDth variable of the input file is selected for above test	37-40
IUT	Fortran serial number referring to output data ($\neq$ 13)	41-44
NBL	Number of blocks in the logical output record	45-48
NEC	Number of words in one block of the logical output record	49-52
MLRECL	1 = Maximum = 1820, 2 = NBL*NEC	53-56
NRUNS	0 = no further run, >0 one further run	57-60
NPRINT	0 = the last output logical record is printed >0 = all logical records are printed	61-64
NEWCØ	0 = subroutine NEWCØD not used. >0 resequencing and spacing for new codes (Form G)	65-68
NSPEC	Serial number of the data set (0 = specification from previous run is kept). (Do not code 0 if no previous run)	69-72
NDFIL	End of file at the end of this run: 0 = no, 1 = yes	73-76
MØDCØR2	1 = Print erroneous records, 2 = print erroneous codes	77-80

Revised 12.7.79

RECØDE	DATE
1+N*2 cards	STUDY
FORM B (continued)	RUN

Run cards (2 cards per run): Card 2.2			Col.
LEND	All variables in the last record contain LEND		1-4
INUTER	1 = Accept, 2 = Reject erroneous records from the recoded output file, only with MODCOR2 = 2		5-8
IER1	Fortran serial number of the file of rejected records (new codes, new sequence), (≠13), with MØDCØR2 = 2 only. 0 = no storage of rejected records		9-12
IER2	Fortran serial number of the file of rejected records (old codes), MØDCØR2 = 2. 0 = no storage and printing of rejected records(≠13)		13-16
LSTBLK	0 = last block not printed, 1 = last block printed (with MØDCØR2 = 1)		17-20

Revised 5.7.71

RECODE	DATE:
3 Parameter Cards	STUDY:
FORM B2 (optional if MØDCØR2 = 2)	RUN

Specification for the revision of codes — Card 1 Col

Card 1: Identification of 1 record: One record can be identified by at most 10 variables, such as village-, household-, individual-serial-number.

a) Indicate how many variables are used for the identification of 1 record.

b) Specify the serial number of each variable. Start preferably with the major identification.

a) Number of variables used for the identification of 1 record: NVAID [ ][ ][ ][ ] 1-4

b) Serial number of each variable used for the record identification:

	5				9				13				17				21				
NIDRC(10)																					5-24
																					25-44
	25				29				33				37				41				

Card 2: Identification of 1 record (text): Each identification is described by a short text. Each identification, except the last, is followed by a comma. The sequence is the same as in card 1: — Card 2 Col

IDRC(20)																					1-20
																					21-40
	1	2	3	4	5	6	7	8	9	0	1	2	3	4	5	6	7	8	9	0	
																					41-60
																					61-80

Description of the variables in 1 record: For each variable one punched card is prepared. Columns 1-28 contain identifications, such as name, page of the questionnaire, column assignment, etc., in the convenient sequence. The programs store at most 1000 descriptions in the array NAMVAR(7,1000).

Card 3: Identification of 1 variable in 1 record: Each identification is described by a short text. Each identification, except the last, is followed by a comma. The sequence is the same as under "description of the variables in 1 record": — Card 3 Col

IDVA(20)																					1-20
																					21-40
	1	2	3	4	5	6	7	8	9	0	1	2	3	4	5	6	7	8	9	0	
																					41-60
																					61-80

Revised 12.7.79

RECODE
0-630 PARAMETER CARDS
FORM C

DATE
STUDY
RUN

0-30 CARDS:1 CARD PER VARIABLE OF 1 CHARACTER
FOR EACH VARIABLE 13 NEW CODE VALUES

CARD	NAME OF VARIABLE	SER.CARD ITM	NO C OL	*	OLD CODE VALUES BL	0	1	2	3	4	5	6	7	8	9	-	&	*
1			1															
2			1															
3			1															
4			1															
5			1															
6			1															
7			1															
8			1															
9			1															
10			1															
11			1															
12			1															
13			1															
14			1															
15			1															
16			1															
17			1															
18			1															
19			1															
20			1															
21			1															
22			1															
23			1															
24			1															
25			1															
26			1															
27			1															
28			1															
29			1															
30			1															

COLUMN 0 1 2 3
123 45 6 78 90 12 34 56 78 90 12 34 56 78 90 12 34

* UNACCEPTABLE NEW CODE VALUE
** CATEGORY OF UNITS OF INFORMATION (FIRST SEQUENCE NUMBER)

Revised 12.7.79

RECODE
0-40*3 Parameter Cards
Form D

DATE:
STUDY:
RUN:

3 Cards per variable — One variable of two characters: 103 new code values

Name of the variable:

NN Item serial number (Form A): Col 1-3

NOCOL** Conversion mode <02 bb → 00 - 99 ; 12 xx → xx> : Col 4-5

Category of units of information (1st seq. number): Col 6

Card 1 — Col
1-3
2 — 4-5
6
Old code blank — 7-8

Old code (Unit)

Old code (tenth)	0 9	0 0	1 1	1 2	2 3	2 4	3 5	3 6	4 7	4 8	5 9	5 0	6 1	6 2	7 3	7 4	8 5	8 6	9 7	9 8	
00																					9-28
10																					29-48
20																					49-68
30													1	2	3	4	5	6	7	8	69-80
30	9	0	1	2	3	4	5	6	7	8	9	0									1-8
											Card 2										
40																					9-28
50																					29-48
60																					49-68
70													1	2	3	4	5	6	7	8	69-80
70	9	0	1	2	3	4	5	6	7	8	9	0									1-8
											Card 3										
80																					9-28
90																					29-48
	9	0	1	2	3	4	5	6	7	8	9	0	1	2	3	4	5	6	7	8	

Old code

	&		-		*	
						49-54
49	50	51	52	53	54	

* Unacceptable new code value.

** NOCOL = 02 : The old code remains unchanged if the new code is blank (or zero). 1-99 cannot be converted to zero.

NOCOL = 12 : The old code is converted to zero if the new code is blank (or zero).

REVISED 16.12.75

RECODE
0-20 PARAMETER CARDS
FORM E

DATE:
STUDY:
RUN:

VARIABLES OF 3 OR MORE CHARACTERS, VALUE OF THE NEW CODES

CARD	NAME OF VARIABLE	SERIAL OF CARD ITEM	NO. OF COLUMNS	CATEGORY	OLD CODE: BLANK	OLD CODE: PUNCH 12 (&)	OLD CODE: PUNCH 11 (-)	NEW CODE *: UNACCEPTABLE VALUE	NEW CODE *: ACCEPTABLE VALUES: RANGE LOWER	NEW CODE *: ACCEPTABLE VALUES: RANGE UPPER	NEW CODE *: ACCEPTABLE VALUES: SINGLE 1	NEW CODE *: ACCEPTABLE VALUES: SINGLE 2
PUNCH IN COLUMN		0 123	45	6	1 789	2 1234 56789012	3 34567890	4 12345678	90123456		5 78901234	
01												
02												
03												
04												
05												
06												
07												
08												
09												
10												
11												
12												
13												
14												
15												
16												
17												
18												
19												
20												
PUNCH IN COLUMN		0 123	45	6	1 789	2 1234 56789012	3 34567890	4 12345678	90123456		5 78901234	

* IF RANGE+SINGLE VALUES=0 OR BLANK: NO TEST IS DONE

Revised 30.8.1978

R E C O D E
0-20 Parameter Cards
Form F

Date:
Study:
Run:

Gaps of N columns: $-99 \leq N \leq 999$

CARD	Serial Card Item (NN)			NØCØL		Category	Gap length in bytes		
1					0				
2					0				
3					0				
4					0				
5					0				
6					0				
7					0				
8					0				
9					0				
10					0				
11					0				
12					0				
13					0				
14					0				
15					0				
16					0				
17					0				
18					0				
19					0				
20					0				
COL	1	2	3	4	5	6	7	8	9

R E C O D E	Date:
0-15 Parameter cards	Study:
FORM G	Run:

New item sequence in the output record:

Old item serial numbers for the new consecutive item serial numbers 1-300

1	2	3	4	5	6	7	8	9	0	1	2	3	4	5	6	7	8	9	0	Col
																				1-20
																				21-40
																				41-60
																				61-80

1	2	3	4	5	6	7	8	9	0	1	2	3	4	5	6	7	8	9	0	
																				1-20
																				21-40
																				41-60
																				61-80

1	2	3	4	5	6	7	8	9	0	1	2	3	4	5	6	7	8	9	0	
																				1-20
																				21-40
																				41-60
																				61-80

1	2	3	4	5	6	7	8	9	0	1	2	3	4	5	6	7	8	9	0	
																				1-20
																				21-40
																				41-60
																				61-80

1	2	3	4	5	6	7	8	9	0	1	2	3	4	5	6	7	8	9	0	
																				1-20
																				21-40
																				41-60
																				61-80

Revised 11.01.1973

G F I L E

Parameter Card 1 Form A

Date:

Study:

Input File X, Job and Run Specifications:

IN	Fortran input unit number (≠ 3)		1- 4
NBL	Number of sub-blocks		5- 8
LSBL	Number of words in one sub-block		9-12
IFPACK	Packed Yes = 1, No = 0		13-16
NCH	Number of household cards (0,20)		17-20
NTYP	Number of episode - types (0,20)		21-24
NIFIX	Number of individual cards in the household (0 = undetermined)		25-28
NIMAX	Maximal number of individual cards (when this number is variable, otherwise it is equal to NIFIX)		29-32
NWI	Number of words in the individual card excluding the words reserved for identification (Zone X)		33-36
KTYPØ	Position of the card-type variable		37-40
KHSPØ	Position of the household serial number		41-44
NEWFIL	0 = Z File only 1 = Z File and ZZ File 2 = ZZ File only		45-48
IWX	Number of words in the zone of the X or ZZ File		49-52
NSDØ	Serial number of the data set		53-56
R	Intermediate storage for parameter cards 2-N: 1 = on the card reader, R = on unit R (≠ 3)		57-60
NRUNS	0 = no further run, > 0 = one further run		61-64
IUTERR	Fortran unit number: rejected parts of the Z file. If zero no file production (≠ 3)		65-68
S	Store parameter cards 2-N on disk: 0 = NØ, 1 = YES		69-72

Revised 15.12.74

G F I L E — Date:

0-1 Parameter card — Study:

FORM AA (Optional)

WORK FILE ZZ

			Col
IUTZZ	Data set reference number ($\neq$ 3)		1- 4
NBLZZ	Number of sub-blocks in the record		5- 8
LSBLZZ	Number of words in the sub-block LSBLZZ = (LSBL - IWX + IWT): unmodified length IWX = IWT = 0 unmodified zone content		9-12
NTYPZZ	Number of record file images selected from file N		13-16
KTYPZZ	Hierarchy 1, 2, 3 record types (HIE types)		

7	8	9	0	1	2	3	4	5	6	7	8	9	0	1	2	3	4	5	6	
																				17-36
																				37-56
																				57-76

Revised 30.8.78

G F I L E

4 Parameter Cards — Date:

Form B — Study:

INPUT FILE X

Card B1: Number of words in the household card of rank N, excluding the words reserved for identification (N = 0, 20) (zone X+Z)

	1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0	Col
NWH		1-20
		21-40
		41-60
		61-80

Card B2: Number of words in the episode card of episode type N, excluding the words reserved for identification (N = 0, 19) (zone X+Z)

	1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0	
NWE		1-20
		21-40
		41-60
		61-80

Card B3: Number of cards for each individual - episode type (0 = undetermined)

	1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0	
LSEQ		1-20
		21-40
		41-60
		61-80

OUTPUT FILE Z

Card B4:

IUT	Fortran output unit number ($\neq 3$)		1- 4
MWB	Maximal number of words in the Z record		5- 8
IWT	Number of words reserved for identification in the Z zone		9-12
NREL	Number of relations (0 - 300)		13-16
NNEWV	Number of new variables (0 - 10)		17-20

Old position in the X file of the Ith zone word of the Z file (if IWT $\neq$ 0)

	1 2 3 4 5 6 7 8 9 0 1 2 3 4 5 6 7 8 9 0	
KPO (I)		21-40
		41-60

NNEWF	Number of new functions (0 - 200)		61-64

Revised 30.8.78

G F I L E, FORM C — Date:

0 - 300 Parameter cards — Study:

OUTPUT FILE Z

			Col
KREL	Relation between cards, KREL=100*I+K, I=1,12= HH,II,EE,IE+, E1E2+, IE1+, ZH,ZI,ZE,IEIE,HI,HIE + K=01=CARD1CARD2,CARD2CARD3,..... 02=CARD1CARD2,CARD3CARD4,..... 11=....,CARD3CARD2,CARD2CARD1. 12=....,CARD4CARD3,CARD2CARD1. 00=CARD1CARD2,CARD1CARD3,..... 10=....,CARD3CARD1,CARD2CARD1.		1- 4
IREL	Relation between variables WRITE a linear combination 1 WRITE a table element 2 WRITE (ad hoc program) 3 IMPLICATION: IF W1 is true MOD (W2*100 + W1, 10 000) = IRES (N), NTEST = N 4 CONSISTENCY: IF MOD(W2*100+W1,10000) ≠ IRES (N), NTEST = N 5	W3 = W1 IF W2 = IRES 6 W3=W1 IFW2 >IRES 7 TEST W1=W2 8 W3=number of cards with W2=IRES 9 W3=number of cards with IRES(I) ≤ W2 < IRES(I+1); NTEST=2*N 10 RECODE A 2 DIGIT VARIABLE W3=NEWV(UNITS,TENTH,NWR) 11	5- 8
NTEST	Number of tests for this relation	TEST W2≥W1 12	9-12
KCAR1	Household-individual-episode card type 1*		13-16
KCAR2	2*		17-20
KCAR3	3*		21-24
KVAR1	Variable (or word) serial number 1*		25-28
KVAR2	2*		29-32
KVAR	3*		33-36
KCOL1	Column number** 1*		37-40
KCOL2	2*		41-44
KCOL3	3*		45-48
KMOD1	Modules** 1*		49-52
KMOD2	2*		53-56
KMOD3	3*		57-60

* The index 1-3 refers to the words 1, 2 and 3 which are the terms of the relation.
** For packed data only.

G F I L E
0 - 25 Parameter Cards
Form D

Date:
Study:
Serial number:

Output File Z

0 - 500 Test results in the sequence of tests and relations

IRES Card	1	2	3	4	5	6	7	8	9	0	1	2	3	4	5	6	7	8	9	0	Col
1																					1-20
																					21-40
																					41-60
																					61-80
	1	2	3	4	5	6	7	8	9	0	1	2	3	4	5	6	7	8	9	0	
2																					1-20
																					21-40
																					41-60
																					61-80
	1	2	3	4	5	6	7	8	9	0	1	2	3	4	5	6	7	8	9	0	
3																					1-20
																					21-40
																					41-60
																					61-80
	1	2	3	4	5	6	7	8	9	0	1	2	3	4	5	6	7	8	9	0	
4																					1-20
																					21-40
																					41-60
																					61-80
	1	2	3	4	5	6	7	8	9	0	1	2	3	4	5	6	7	8	9	0	
5																					1-20
																					21-40
																					41-60
																					61-80
	1	2	3	4	5	6	7	8	9	0	1	2	3	4	5	6	7	8	9	0	
6																					1-20
																					21-40
																					41-60
																					61-80

G FILE

0-100 Parameter Cards

Form E

Date
Study
Serial number of table(s)

Output File Z

NEWV (WORD2, WORD1, SERIAL OF TABLE)

SERIAL OF RELATION

WORD 2

CARD (WORD1)

Col

1-40

SERIAL OF RELATION

WORD 2

1-40

Revised 30.08.78

G FILE

0-200 Parameter Cards

FORM E2

Date
Study
Serial number

COLUMN						CARD
1 4	8	12	16	20	24 28	
*FSN	A	B	C	D	E F	
						1
						2
						3
						4
						5
						6
						7
						8
						9
						10
						11
						12
						13
						14
						15
						16
						17
						18
						19
						20
						21
						22
						23
						24
						25
1 4	8	12	16	20	24 28	Col

*Abbreviations:. FSN function serial number; A-F constants; X new variable;
Y1, Y2 old/intermediate variables (WORD1, WORD2)

LINEAR FUNCTIONS: 1. X=Y1: 2. X=A*Y1: 3. X=(Y1/B)*A; 4. X=(Y1/B)*A + (Y2/D)*C
5. X=(Y1/B)* A + (Y2/D) * C+E; 6. X = A* Y1+ B* Y2; 7. X = A
12. If A,F = -999 it is replaced by NSDS and function 9 is calculated.

TRANSFERS: 8. X=Y1 if Y2≤B, X=A otherwise; 9. X=Y1 if Y2≠B, X=A otherwise.

TEST IMPLICATION between 2 propostions, P1 and P2:

10. P1 (A < I1≤B if C=1 then Y1∈ I1)
if C=0 then Y1∉ I1) A,B,C,D,E, and F being defined, if P1 does not imply P2: error message.

P2 (D < I2≤E if F=1 then Y2∈ I2)
if F=0 then Y2∉ I2)

PERCENTAGE: 11. X=Y1/Y2·1000 with X=0 if Y2=0.

REVISED 30.08.1978

GTAB
PARAMETER CARD 1
FORM A

DATE
STUDY

		COL
MAXTAB	NUMBER OF TABLES TO BE PROCESSED	01-04
NRUN	NOT USED	05-08
NFV	NUMBER OF INDEPENDANT VARIABLES	09-12
D	FORTRAN DATA SET REFERENCE NUMBER FOR INPUT DATA (NOT 9)	13-16
NSBL	NUMBER OF SUBBLOCKS IN THE RECORD OF DATA SET D	17-20
LSBL	LENGTH OF SUBBLOCK IN WORDS	21-24
IP	=1 FOR PACKED DATA,= BLANK OR 0 OTHERWISE	25-28
IPST	=1 READ AND STORE PACKING PARAMETER TABLE, =0 OR BLANK OTHERWISE	29-32
KCH	NUMBER OF CATEGORIES IN HIERARCHY 1 (0-20)	33-36
KTYP	NUMBER OF CATEGORIES IN HIERARCHY 3	37-40
NWI	NUMBER OF WORDS IN THE CATEGORY OF HIERARCHY 2 EXCLUDING WORDS RESERVED FOR IDENTIFICATION (ZONE X)	41-44
IWT	NUMBER OF WORDS IN THE ZONE Z	45-48
MSNFIL	NUMBER OF SUBFILES IN D	49-52
S	FORTRAN DATA SET REFERENCE NUMBER FOR DIRECT ACCESS TO TABULATION VARIABLES =0 READ DESCRIPTIONS AFTER TABLE SPECIFICATION (CARD READER) =9 STORE DESCRIPTIONS ON THE DISK	53-56
IWX	NUMBER OF WORDS IN THE INPUT ZONE X OR ZZ	57-60
IERROR	=1 PRINT THE LIST OF DESCRIPTIONS OF THE VARIABLES =0 NO PRINT	61-64
LANG	=0 OR BLANK THE LANGUAGE USED FOR THE TITLES IS ENGLISH =1 THE LANGUAGE USED FOR THE TITLES IS FRENCH	65-68
IN	FORTRAN DATA SET REFERENCE NUMBER FOR THE INPUT FILE OF TABULATION VARIABLES: =1 CARD READER; NOT 9 TAPE	69-72
RECFOR	=0 INPUT DATA D IS A FILE OF TYPE Z =1 INPUT DATA D IS A FILE OF TYPE ZZ	73-76

GTAB

Parameter Cards — Date:

FORM B — Study:

PROJECT CHARACTERISTICS 2

Col

Card B1: PNAM(20): 20 Files names for 20 data sets

1	2	3	4	5	6	7	8	9	0	1	2	3	4	5	6	7	8	9	0	
																				1-20
																				21-40
																				41-60
																				61-80

Card B2: NWE: Number of words in the hierarchy 3 cards of rank N (E type), excluding the words reserved for identification (zone)

1	2	3	4	5	6	7	8	9	0	1	2	3	4	5	6	7	8	9	0	
																				1-20
																				21-40
																				41-60
																				61-80

Card B3: KWH: Number of words in the hierarchy 1 card of rank N (H type), excluding the words reserved for identification (zone)

1	2	3	4	5	6	7	8	9	0	1	2	3	4	5	6	7	8	9	0	
																				1-20
																				21-40
																				41-60
																				61-80

Revised 15.12.74

GTAB — Date
1-2 cards: table definition
FORM C — Study

Standard code (underlined), title MST value are written by the program when boxes are left blank. Necessary coding in card 2 columns 55-80

Table type:	CARD 1		COL.
0 = frequency distribution, 1 = frequency distribution + marginal totals (SN), 2 = code 1 + marginal per thousand (SN/TOT), 3 = code 0-2 + cumulative 0/00 distributions, TOT, SX, SX2, XBAR, VARX, SD, 4 = code 0-2 + 0/00 relative to marginal totals, 5 = rates 6 = TOT, SX, SX2, XBAR, VARX, SD, 7,8 = control table (subtable C): RATE, MEAN.	ICALCP		1
Table presentation:			
Parameters: 0 = none, 1 = SN, 2 = SN/TOT	ISHØPA		2
Where: 1 = right margin, 2 = last line 3 = both places, 4 = SN as subtables (+ code 3) 5 = SN as subtables (+ code 1)	ISHØPL		3
Next tables are a repetition of this table, after new definitions (FORM G) 00 or 01 = no repitition	MAXREP		4- 5
This table consists of ISMBL column groups, 00 = 01 = no	ISMBL		6- 7
Lines with 0 only: 0 = printed, 1 = not printed	MODU		8- 9
Number of variables directing the classification:			
as column headings code 1, 2, 3	NCH		10
as line headings code 1	NLH		11
as subtable headings code 0-4	NSH		12
Variable used as measurment code 0-1	NMH		13
Table printout:			
Number of characters per column 5, 6	KCARA		14-15
No print of KØLSØ first columns 0	KØLSØ		16-17
No print of these columns which follow KØLØN 0	KØLØN		18-19

Title

20	24	28	32	36	40	44	47	
								20-47

48	52	56	60	64	68	72	75	
								48-75

Title: 0 = no, 1 = on this card only, 2 = on following card also, 3-9 words for table serial number (7)	NTC		76
Subtitle: 0 = yes, 1 = yes for 1st subtable, 2 = no	NST		77
Column headings: 0 = yes, 1 = no, 2 = yes for 1st subtable	NCT		78
Title: 0 = on the same page, 1 = new page (A4 format); add 2 to this code for A3 format; (3)	MTC		79
Number of subtables per page	MST		80

Title (continuation of the title text, col. 20-75 of card 1) — CARD 2 (if NTC ≥ 2) — COL.

1	5	9	13	17	21	25	28	
								1-28

29	33	37	41	45	49	53	56	
								29-56

* *

Variables should be defined in the sequence NCH, NLH, NSH, NMH
Variable serial numbers (for direct access only)

	57	60	63	66	69	72	74	75	78	80	
NVAC											57-80

** Write ST in col. 55-56 if no form F is used (standard table).

Revised 30.8.78

GTAB: 1-14 cards: definition of a variable: FORM D

DATE STUDY

CARD 1: VANAM — Col.

NAME OF VARIABLE: 1 5 9 13 17 21 25 28 — 1-23

NUMBER OF 1/2 WORDS IN THIS NAME — LGTHVA — 29-32

STORAGE SERIAL NUMBER OF THE VARIABLE, CARD TYPE — NSTV 33-36 NSTC — 37-40

CARD 2 — Col.

This variable is used for: 1=measurement, 2=classification, 3=1+2
4=initialization of n classifications with n variables specified in FORM E
5=initialization of C_n^n classifications with n binary (yes=2, no=1) variables specified in FORM E, 6=the value NEWVAR (col 1-4, FORM E) will not indicate the position of this new variable in the table
8: See KREAD=8, FORM F, GTAB.

ITYP — 1

Recoding the measurement:
1) Add MED to the measurement; 2) If MED=99: MES/10

MED — 2-3

Recoding the classification variable:
Final codes are consecutive integers, starting at

LU* — 4-5

Following transformations are processed in the sequence 1-5 on option:

1. Addition of a constant: lowest new code 1 corresponds to old code — ICCL — 6-7
3. Replacement: number of 1 digit codes to be replaced — NOC — 8-9

   List of 1 digit old codes (ascending order) — MCOD — 10 15 — 10-19
   List of corresponding new codes — NCOD — 20 25 — 20-29

4. Codes are grouped into classes, classes into types: number of types (code 1-3) — ICG — 30

   if ICG≠0, lowest code in each class - type — IL1 — 31 35 39 42 — 31-42
   increment in each class type — ICR — 43 46 49 — 43-51
   lowest code in last class — IL1 — 52 55 — 52-55

2. Module transformation: codes 1-5, 6-10, ... replaced by 1-5, 1.5, ... (MT=5) — MT — 56-57
   IF (MT(I) = 99) MT (I) = 100

Subtitle element or column or line heading (text element)

Number of characters reserved for 1 text element: 4,8,12 or 16 — LCØD — 58-59
Number of text elements for this variable — NNAM — 60-61
Word serial number in the record (packed input data only) — MWØRD — 62-63
Column serial number in above word (packed input data only) — MCØLU — 64-65
Variable serial number — NVACA — 66-68
5. Add a constant — KCCL — 69-70

Next card GNAM
1 5 9 13 17 21 25 28 — 1-28
29 33 37 41 45 49 53 56 — 29-56
57 61 65 69 73 77 80 — 57-80

Next card GNAM
1 5 9 13 17 21 25 28 — 1-28
29 33 37 41 45 49 53 56 — 29-56
57 61 65 69 73 77 80 — 57-80

Next card GNAM
1 5 9 13 17 21 25 28 — 1-28
29 33 37 41 45 49 53 56 — 29-56
57 61 65 69 73 77 80 — 57-80

* LU participates to the calculation of the dimensions of the table, after optional transformations 1-5.

GTAB

0-4 cards: secondary variables

FORM E

Date:

Study:

Initialization of n or C_n^n classifications with n variables

(Form a new variable with n variables)

CARD 1

Col

NEWVAR: Serial number (code 1-8) of the variable 'List of names of the associated variables' in the table variables — 1-4

LISMAX or MAXIMO: Number of variables forming the new variable — 5-8

MWORT or MPARO (≤60): Serial number of variables in the work file (FORM A, RECODE)*

1	2	3	4	5	6	7	8	9	0	1	2	3	4	5	6	7	8	9	0	
																				9-20
																				21-40
																				41-60
																				61-80

(CARD 2)

1	2	3	4	5	6	7	8	9	0	1	2	3	4	5	6	7	8	9	0	
																				1-20
																				21-40
																				41-60
																				61-80

(CARD 3)

1	2	3	4	5	6	7	8	9	0	1	2	3	4	5	6	7	8	9	0	
																				1-20
																				21-40
																				41-60
																				61-80

(CARD 4)

1	2	3	4	5	6	7	8	
								1-8

* If packed data are used and if a parameter table is not used, MWORT or MPARO should be followed by MKØLØ and MBASE or MPILA and MBASO, these two words indicating column and base.

Revised 30.8.1978

GTAB
2 cards (optional)
FORM F

Date

Study

CARD 1

Number of classifications or subtables (code 1-3) — NEL — 1

If only 1 classification is used do not code columns B and C:
Unit in subtable(s) A,B,C
1=of hierarchy 1; 2=of hierarchy 2; 3=of hierarchy 3;
4=sum of x (in B only); 5=A/B (in C only); 6=B/A (in C only)
The variables are read in:
K=all records of 1 type in hierarchy K (K=1 to 3)
4=1/several hierarchy(ies), 1/several type(s), selected/all record(s)

	A	B	C	
INCEL				2-4
LIR				5-7

Control table of type 0, 1 or 2 (Form C): If NVS≠0 the IVAR variables (cols 45-52) specified in the column of rank NVS will be read as 1 when INCEL(2) (column 3) is not blank — NVS — 8

IS1, IS2, IS3, IS4, IS5, IS6, IS7 — 9 11 13 15 17 19 21 22 — 9-22

Localization and reading of variables and classifications:

Variables defined by Forms D (in the sequence NCH, NLH, NSH, NMH) — first variable(s)

Classification(s) or Subtable(s)

		23	25	27	29	31	33	35	37	39	41	43	44	
Record type	KCAR									A	B	C		23-44
		45	46	47	48	49	50	51	52	53	54	55		
Number of variables	IVAR													45-55
		56	57	58	59	60	61	62	63	64	65	66		
Reading and classification are executed by cycle(s): indices=1 when cycle unused	KE1													56-66
		67	68	69	70	71	72	73	74	75	76	77		
	KE2													67-77

Cycles on hierarchy 3:
KE1: initial serial of the record (0=last)
KE2: final serial (0=last record)
KE3: increment

Cycles on hierarchy 2:
I1: initial serial of the record
I2: final serial (0=last record)

CARD 2

		1	2	3	4	5	6	7	8	9	10	11	
	KE3												1-11
		12	13	14	15	16	17	18	19	20	21	22	
	I1												12-22
		23	24	25	26	27	28	29	30	31	32	33	
	I2												23-33

KREAD: The variable(s) is/are used for: 1=measurement and classification; 2=classification only; 3=classification according to the number of records with code 2 for this variable; 4=initialization of C_n^n classifications with n binary variables specified in FORM E; 5=initilization of n classifications with n variables specified in FORM E; 6=if this variable is on a hierarchy3 missing record previous value of the variable is kept; 8=new variable is: 1(yes) if at least 1 of the variables of form E is recoded 1
2(no) if all the N variables of form E are recoded 2
3(unknown) otherwise

	34	35	36	37	38	39	40	41	
KREAD									34-41

GTAB — Date

1-2 cards — Study

FORM G (optional)

Form G should be used when the previous table is repeated. 1-7 variables of the previous table should be read now in other locations of the record (example: results of a next survey of a longitudinal study).

Col.

Card 1

For each variable which should be read elsewhere indicate the value of:
MVSN: Serial number of the variable in the table
MMØT, MCØLØ, MPILI: New address of the variable in the record

Modification	1					1-7
"	2					8-14
"	3					15-21
"	4					22-28
"	5					29-35
"	6					36-42
"	7					43-49

New title for this table

50 1 2 3 4 5 6 7 8 9 60 1 2 3 4 5 6 7 8 9 70 1 2 3 4 5 6 7

50-77

NTC Title: 0 = no, 1 = on this card only, 2 = on following card also — 78

MTC Title: 0 = on the same page, 1 = new page — 79

Card 2 (optional): Title continuation

1 2 3 4 5 6 7 8 9 10 1 2 3 4 5 6 7 8 9 20 1 2 3 4 5 6 7 8

1-28

29 30 1 2 3 4 5 6 7 8 9 40 1 2 3 4 5 6 7 8 9 50 1 2 3 4 5 6

29-56

= = =