LISTS: A SIMPLE MIXED (MANUAL/COMPUTER) PROCEDURE FOR MAINTAINING FORMATTED LISTS AND INDEXES

by

Ronald W. Helms

Department of Biostatistics
University of North Carolina at Chapel Hill

Institute of Statistics Mimeo Series No. 994

April 1975

LISTS: A SIMPLE MIXED (MANUAL/COMPUTER) PROCEDURE FOR MAINTAINING FORMATTED LISTS AND INDEXES

By Ronald W. Helms

1. INTRODUCTION

LISTS is a procedure which contains three components:

(a) A simple form for entering data;

(b) A card data file with manual updating procedures or, for large files, a card-image sequential data file on tape or disk maintained by standard update programs; and

(c) A computer program which produces formatted listings from the data file.

The procedure is simple to operate; once the procedure has been set up secretaries with no data processing background can be quickly trained to operate it. No programming is needed to set up or operate the procedure. The procedure is very flexible: a wide variety of list formats can be produced very simply. LISTS can be used to maintain such diverse files as personnel directories, mailing lists, low-turnover inventories, listings of hierarchal structures, etc.

Section 2 of this report contains a description of the LISTS procedure and the information needed to use the various components. Two simple examples of the application of the procedure are given in Section 3 and 4. The first example illustrates the maintenance of a hierarchally structured file index. Section 5 is an appendix containing some technical specifications.

Persons with access to the Triangle Universities Computation Center may use the LISTS program directly from a disk file, as described in Section 2.4. Others may obtain a copy of the program deck by contacting the author.

2. THE LISTS PROCEDURE

2.1. Description of the Algorithm and Output

The LISTS computer program is basically a computer program for listing the contents of cards. The cards must have the special format described below. Each card contains five fields named INDEX, INDENT, CAR_CTL (abbreviation of "carriage control"), LINE_NO (an abbreviation of "line number"), and TEXT. The purpose of the LISTS program is to print the information in the INDEX and TEXT fields. The INDENT field permits the user to control the horizontal location where the TEXT field is printed. The CAR_CTL field permits the user to control the vertical spacing between lines, and the LINE_NO field controls the sequence of lines within a block of lines, as, for example, the lines within an address.

The basic steps of the operation of the program are as follows:

1. Read a card (or card-image).
2. Skip the number of lines indicated by the CAR_CTL field. This produces CAR_CTL-1 blank lines on the output. If CAR_CTL = 'P' (the letter P), the printer is skipped to the top of a new page.
3. Shift the TEXT so the first printing position will be in position 12 + INDENT * 4. This has an effect similar to tabulating INDENT times on a typewriter, where each indention represents 4 print positions.
4. If LINE_NO is blank or 0 (zero), the INDEX field is moved to positions 1-10 of the same line as the TEXT field.
5. The line is printed and the program returns to step (1).

There are several complications not treated in the basic algorithm described above. For example, if the CAR_CTL field contains a 'T', the next field is treated as a title and is printed at the top of each page until a subsequent card is encountered with a CAR_CTL field containing a 'T'. Thus, several titles may be used in one run.

The header printed at the top of each page includes the title, if any, the date of the run, time of day, and page number. The page numbering is reset to 1 each time the title changes.

Since each card contains carriage control and indention specifications the user has considerable flexibility in arranging format of the printed output. Illustrative examples are given in Sections 3 and 4 of this report.

2.2. The LISTS Form

The form which has been designed for the LISTS procedure is illustrated in Figure 1. The form is specifically designed to be printed on card stock the same size as standard data processing cards. It is important that the form be printed on card stock without corner cuts; i.e., the cards should have four square corners. (This is explained in Section 2.3.)

Each form contains information to be punched into a block of up to 10 cards. A block of cards is a set of cards which contain identical information in the INDEX field. The LISTS form is designed so that the INDEX field is punched into the first card (the card with LINE_NO = 0) and reproduced into all subsequent cards of the block. The use of the INDEX field is illustrated in Sections 3 and 4.

The line numbers (column 13) are preprinted on the LISTS form as 0, 1...,9. Only as many lines as are actually used are punched, i.e., lines with blank TEXT fields (columns 14-80) are not punched.

Each line has a space for inserting the CAR_CTL (carriage control) character (column 12) and a space for the INDENT character (column 11). Usually every card (line) in a block will have the same INDENT value.

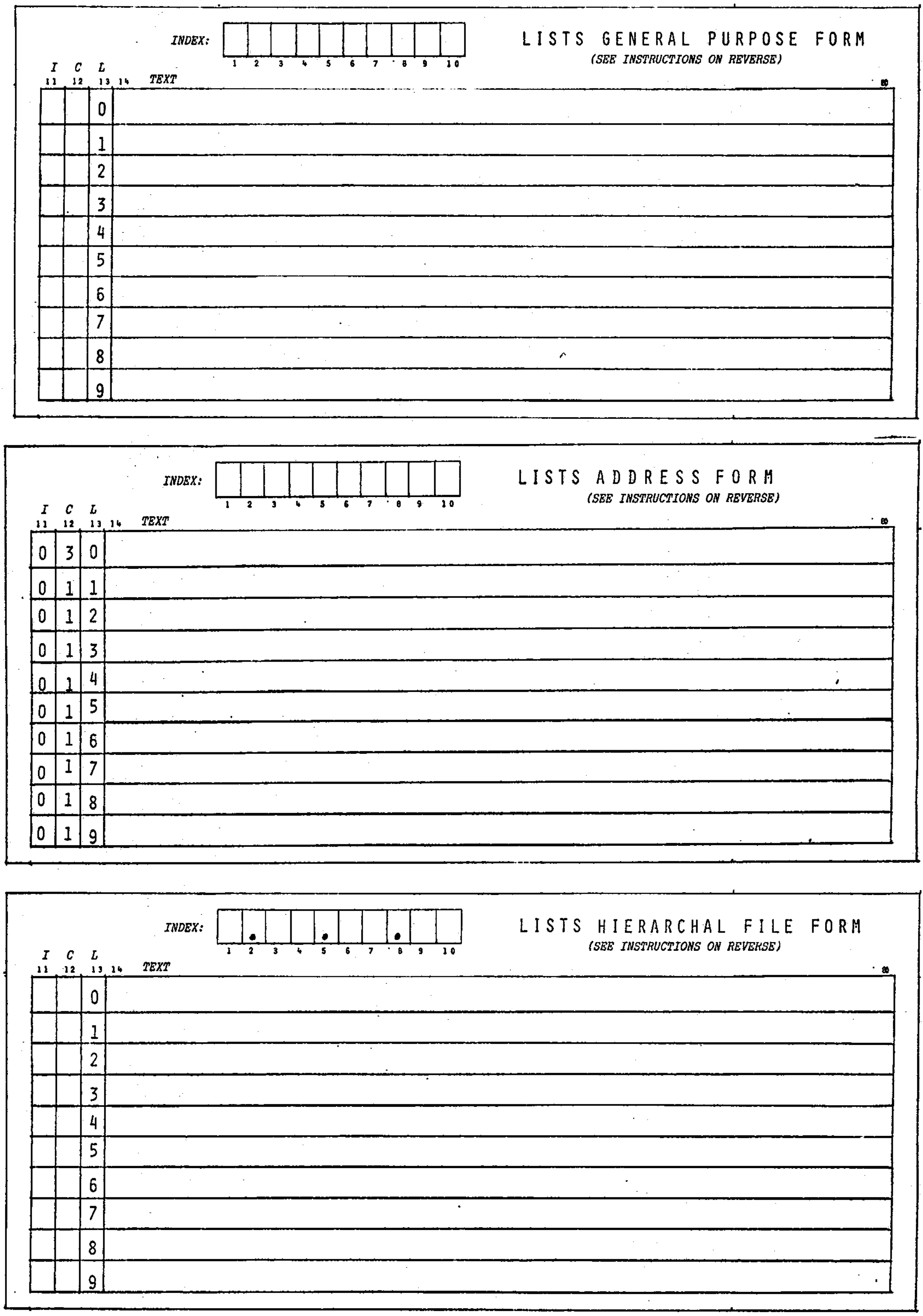
The CAR_CTL character for the first line of a block controls the number of spaces between the block and the previous block while the CAR_CTL characters of subsequent cards in a block control within-block line spacing. Thus, the CAR_CTL character of the first line usually has a value such as 2, 3, or 4 to separate this block from the preceeding block by one or more blank lines, while the subsequent CAR_CTL characters are usually blank (to give single spacing within the block).

Because of the considerations discussed in the previous two paragraphs, one typically fills out the following "control fields" of the LISTS form:

1. The INDEX field;
2. The CAR_CTL field of the first line (leaving blank the CAR-CTL fields of subsequent lines); and
3. The INDENT field of each line, using the same value in each line.

The TEXT fields are filled in with the appropriate information (e.g., name and address).

A "camera-ready" copy of the LISTS form, suitable for reproduction,



INDEX: □□□□□□□□□□ 1 2 3 4 5 6 7 8 9 10

LISTS GENERAL PURPOSE FORM

(SEE INSTRUCTIONS ON REVERSE)

I 11	C 12	L 13	14 TEXT 80
		0	
		1	
		2	
		3	
		4	
		5	
		6	
		7	
		8	
		9	

INDEX: □□□□□□□□□□ 1 2 3 4 5 6 7 8 9 10

LISTS ADDRESS FORM

(SEE INSTRUCTIONS ON REVERSE)

I 11	C 12	L 13	14 TEXT 80
0	3	0	
0	1	1	
0	1	2	
0	1	3	
0	1	4	
0	1	5	
0	1	6	
0	1	7	
0	1	8	
0	1	9	

INDEX: □ □. □ □ □. □ □ □. □ □ 1 2 3 4 5 6 7 8 9 10

LISTS HIERARCHAL FILE FORM

(SEE INSTRUCTIONS ON REVEKSE)

I 11	C 12	L 13	14 TEXT 80
		0	
		1	
		2	
		3	
		4	
		5	
		6	
		7	
		8	
		9	

FIGURE 1. The LISTS General Purpose Form and two special purpose forms used in the examples. The forms are shown approximately 3/4 actual size.

is available from the author.

2.3. Maintaining a LISTS File

Since the LISTS procedure was designed to be operated by secretarial or clerical personnel with no data processing training, file maintenance is necessarily an almost trivial manual operation.

A file is initially created on data processing cards in the format specified by the LISTS form. Different files are maintained as separate card decks, but may be combined for a LISTS computer run. Within each card deck the blocks are ordered according to the INDEX field. (A block is a set of cards with the same INDEX field value.) For example, in an address file application the INDEX field typically contains last name and initials; the file would be maintained in alphabetical order according to last name and initials, i.e., according to the INDEX field.

Within a block the cards are arranged according to line number (column 13).

To add a block of one or more cards to the file one fills out the LISTS card and inserts it in the card deck at the proper position for the block. Additions are made in this manner until it is desired to make a LISTS run to list the file. Before the LISTS run, the secretary takes the card deck to a keypunch, (1) removes a LISTS form (marking the place with another card inserted vertically), (2) keypunches the information from the LISTS form into the appropriate number of cards, (3) proofreads the keypunched cards, comparing them with the LISTS form, and (4) inserts the punched cards into the deck at the proper place. This simple procedure is repeated for each LISTS form. The LISTS forms are easily located in the deck because both the top-left and top-right corners are square, not cut. The punched cards of the file are cards with either the top-left or top-right corners cut, so a corner of each LISTS form is easily seen.

To delete a block from a LISTS file one simply removes the corresponding cards.

To correct an error in a card in the LISTS file one simply fills out that line of a LISTS form corresponding to the erroneous card and replaces the erroneous card by the LISTS form. Before the next LISTS run, the secretary will keypunch the correction and replace the LISTS card with the corrected card. An alternative method of error correction is to write a note describing the correction on the erroneous card and turn the card around backwards in the punched deck. The corners of such cards will "stick out" and the corrections will be punched when the new LISTS forms are punched.

2.4. Making a LISTS Computer Run

The control cards required to execute the LISTS program will be set up when the LISTS file is originally set up. No changes need be made to the control cards on subsequent runs. After punching new LISTS data and making corrections, the secretary simply submits the already set-up deck for execution by the computer.

The following control cards and deck arrangement can be used to execute the LISTS program at the Triangle Universities Computation Center.

```
//    jobcard
//*PW=password
//JOBLIB  DD  DISP=SHR,DSN=UNC.ES.F245E.HELMS.LOADLIB

//    EXEC  PGM=LISTS
//SYSPRINT DD SYSOUT=A,DCB=(RECFM=VBA,LRECL=137,BLKSIZE=141)
//INPUT  DD  *
      (title card for first LISTS file)

     {data cards for first LISTS file}

     (title card for second LISTS file)

     {data cards for second LISTS file}

     (etc)

     /*
```

3. EXAMPLE: USING *LISTS* TO MAINTAIN AN ADDRESS FILE OR MAILING LIST

This example illustrates one of the simplest applications of the LISTS procedure. The objective is to maintain a directory of persons, including names, addresses, and telephone numbers. For example, a coordinating center for a collaborative medical trial may wish to maintain a list of all the personnel at participating centers. Another example might be maintenance of a list of members of committees.

The following paragraphs describe one way of setting up the LISTS procedures for this problem.

The INDEX field. In this example each block (each set of cards with the same INDEX) is the name, address, and telephone number correspoding to one person. The INDEX field is set up such that:

(a) the first entry, beginning in column 1, is the last name (surname).

(b)a blank follows the surname.

(c) the blank is followed by the individual's initials, without spaces or punctuation

(d) if the field, as defined above, exceeds 10 characters, delete characters from the right to obtain a field 10 characters long.

(e) if the result of (d) leaves two or more blocks with the same INDEX field, use only the first 8 characters of the surnames followed by up to two initials in positions 9 and 10.

Name	INDEX Field
Dr. John Albers	ALBERS J
Dr. Charles Ash	ASH C
Dr. Paul S. Bachorik	BACHORIK P
Dr. W. Carl Breckenridge	BRECKENRID
Mr. Ralph A. Carmichael	CARMICHARA
Mr. James Carmichael	CARMICHAJ

Remember: the INDEX field is used only to control the order of the address in the file; the TEXT field contains the full name and title (Mrs., Dr., etc.) and when the file is printed the full name and title, up to 67 characters, are printed.

The INDENT field. Usually indention is not used in address lists. In this example the INDENT field is always left blank (no indention) or punched with a zero.

The CAR_CTL field. Typically one wishes to have several blank spaces between addresses but single spacing within the address block. This is achieved by using:

(a) A 3 for the CAR_CTL field (column 12) of the first line (line 0) of each name-and-address block; and

(b) A 1 (or blank) for the CAR_CTL field of each succeeding line within the block.

These values result in 2 blank lines between addresses and single spacing within addresses.

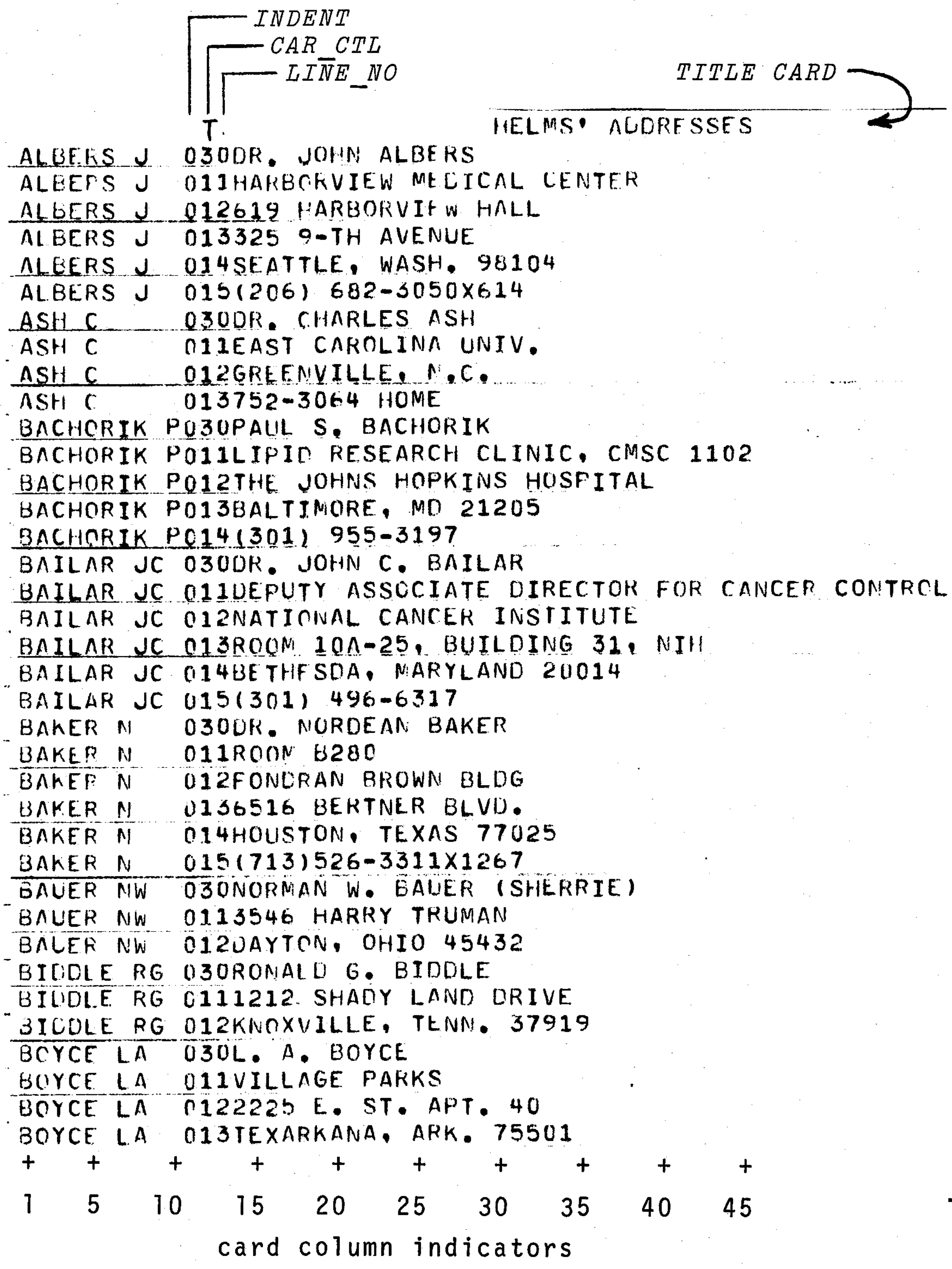
The LINE-NO field. The line numbers are used as indicated on the LISTS form, i.e., the first line of the address (the name line) has line number 0, the second has line number 1, etc. This scheme permits up to 10 lines in a block, which is usually more than adequate. If more than 10 lines are needed, one could use a blank for the first line number and the letters A, B,...,etc. for succeeding line numbers. In this case more than one LISTS form would be required for the block.

The TEXT fields. In the TEXT fields one simply enters the lines of the name, address, and, if desired, the telephone number. Each line can accomodate up to 67 characters of information, including spaces and punctuation.

Example, Figure 2 contains a listing of part of a LISTS card file used for a name-and-address application. Figure 3 is an example of the output printed by the LISTS program.

Note that the first card in Figure 2 is a title card; this is indicated by the 'T' in the CAR_CTL field (column 12.)

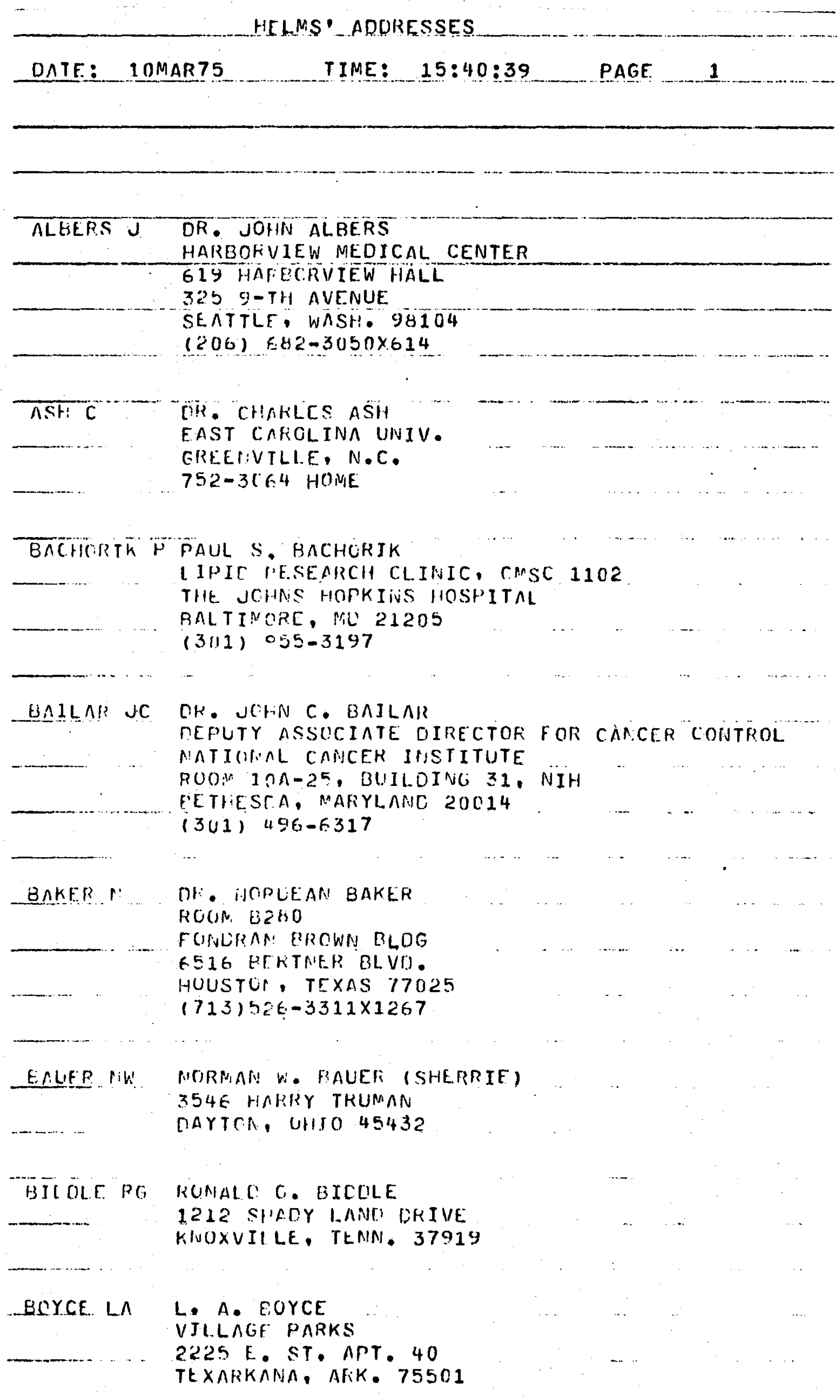
These printouts were made on standard computer output paper. At many installations it is possible to get special paper, such as paper without lines, paper only 8 1/2 inches wide, etc. If special forms are



INDENT
CAR_CTL
LINE_NO
TITLE CARD
card column indicators

```
                              HELMS' ADDRESSES
ALBERS J  030DR. JOHN ALBERS
ALBERS J  011HARBORVIEW MEDICAL CENTER
ALBERS J  012619 HARBORVIEW HALL
ALBERS J  013325 9-TH AVENUE
ALBERS J  014SEATTLE, WASH. 98104
ALBERS J  015(206) 682-3050X614
ASH C     030DR. CHARLES ASH
ASH C     011EAST CAROLINA UNIV.
ASH C     012GREENVILLE, N.C.
ASH C     013752-3064 HOME
BACHORIK P030PAUL S. BACHORIK
BACHORIK P011LIPID RESEARCH CLINIC, CMSC 1102
BACHORIK P012THE JOHNS HOPKINS HOSPITAL
BACHORIK P013BALTIMORE, MD 21205
BACHORIK P014(301) 955-3197
BAILAR JC 030DR. JOHN C. BAILAR
BAILAR JC 011DEPUTY ASSOCIATE DIRECTOR FOR CANCER CONTROL
BAILAR JC 012NATIONAL CANCER INSTITUTE
BAILAR JC 013ROOM 10A-25, BUILDING 31, NIH
BAILAR JC 014BETHESDA, MARYLAND 20014
BAILAR JC 015(301) 496-6317
BAKER N   030DR. NORDEAN BAKER
BAKER N   011ROOM B280
BAKER N   012FONDRAN BROWN BLDG
BAKER N   0136516 BERTNER BLVD.
BAKER N   014HOUSTON, TEXAS 77025
BAKER N   015(713)526-3311X1267
BAUER NW  030NORMAN W. BAUER (SHERRIE)
BAUER NW  0113546 HARRY TRUMAN
BAUER NW  012DAYTON, OHIO 45432
BIDDLE RG 030RONALD G. BIDDLE
BIDDLE RG 0111212 SHADY LAND DRIVE
BIDDLE RG 012KNOXVILLE, TENN. 37919
BOYCE LA  030L. A. BOYCE
BOYCE LA  011VILLAGE PARKS
BOYCE LA  0122225 E. ST. APT. 40
BOYCE LA  013TEXARKANA, ARK. 75501
+   +    +    +    +    +    +    +    +    +
1   5    10   15   20   25   30   35   40   45
```

FIGURE 2. Part of a LISTS file containing names and addresses



```
                    HELMS' ADDRESSES

 DATE:  10MAR75        TIME:  15:40:39      PAGE    1



 ALBERS J     DR. JOHN ALBERS
              HARBORVIEW MEDICAL CENTER
              619 HARBORVIEW HALL
              325 9-TH AVENUE
              SEATTLE, WASH. 98104
              (206) 682-3050X614

 ASH C        DR. CHARLES ASH
              EAST CAROLINA UNIV.
              GREENVILLE, N.C.
              752-3064 HOME

 BACHORIK P   PAUL S. BACHORIK
              LIPID RESEARCH CLINIC, CMSC 1102
              THE JOHNS HOPKINS HOSPITAL
              BALTIMORE, MD 21205
              (301) 955-3197

 BAILAR JC    DR. JOHN C. BAILAR
              DEPUTY ASSOCIATE DIRECTOR FOR CANCER CONTROL
              NATIONAL CANCER INSTITUTE
              ROOM 10A-25, BUILDING 31, NIH
              BETHESDA, MARYLAND 20014
              (301) 496-6317

 BAKER N      DR. NORDEAN BAKER
              ROOM B280
              FONDRAN BROWN BLDG
              6516 BERTNER BLVD.
              HOUSTON, TEXAS 77025
              (713)526-3311X1267

 BAUER NW     NORMAN W. BAUER (SHERRIE)
              3546 HARRY TRUMAN
              DAYTON, OHIO 45432

 BIDDLE RG    RONALD G. BIDDLE
              1212 SHADY LAND DRIVE
              KNOXVILLE, TENN. 37919

 BOYCE LA     L. A. BOYCE
              VILLAGE PARKS
              2225 E. ST. APT. 40
              TEXARKANA, ARK. 75501
```

FIGURE 3. The LISTS output from the address file shown in Figure 2.

used, be sure the paper is wide enough. The printed output requires

12 + Max Indent x 4 + Max Length

printing spaces, where

Max Indent =	the maximum INDENT value used (0 in this example)
Max Length =	the maximum number of characters actually used in a line of TEXT. Max Length is always $\leq$ 67.

4. EXAMPLE: USING *LISTS* TO MAINTAIN AN INDEX OF A FILE WITH HIERARCHAL STRUCTURE

This application is based upon an office file (such as in maintained in a standard office filing cabinet). The file has a hierarchal structure (explained below). The file is fairly active, with approximately 10-20 additions and deletions made weekly. It is necessary to produce up-to-date listings of the file index, about 20 pages long, approximately once per week. The LISTS procedure is used to maintain the file index and produce listings which reflect the hierarchal structure of the file.

The hierarchal structure of the file can be explained as follows. The file contains a number of major categories, as, for example:

INDEX	MAJOR CATEGORY TITLE
A.	Administrative Matters
C.	Consulting Projects
R.	Research Projects

Within each Major category there are several (regular) categories, as, for example:

INDEX	CATEGORY TITLE
A.01	Personnel
A.02	Academic Matters
A.04	Committees
...	...
C.01	Norfolk (Hampton Roads) Health Information Center
C.02	NCHS Health Examination Survey
...	...

Within each category there may be one or more sub-categories, e.g.:

LEVEL	INDEX	TITLE
(Major cat.)	A.	Administrative Matters
(catergory)	A.01	Personnel
(sub-category)	A.01.01	Computing Group Organization
(sub-category)	A.01.02	Personnel Actions in Progress
(sub-category)	A.01.03	Information on State Classification System
(sub-category)	A.01.04	Personnel Forms

Within each sub-category there may be one or more sub-sub-categories, e.g.:

(major category)	K.	KIDNEY TRANSPLANT HISTOCOMPATIBILITY STUDY
(category)	K.01	CONTRACT AND FINANCIAL
(sub-category)	K.01.01	SDMC RFP'S
(sub-category)	K.01.02	SDMC PROPOSALS
(sub-sub-cat.)	K.01.02.01	ORIGINAL SDMC PROPOSAL
(sub-sub-cat.)	K.01.02.02	SDMC PROPOSAL--YEAR 02
(sub-category)	K.01.03	SDMC CONTRACTS

These examples illustrate the hierachal nature of the filing scheme and the fact that the format of the listing, by use of variable line spacing and indention, is a visual aid to the reader, guiding the eye through the structure of the file.

The following paragraphs describe how a LISTS procedure is set up to maintain an index of the file and produce formatted listings of the index.

The INDEX field. In this case the INDEX field contains the index number (as for example, A., A.01, A.01.01, etc.) of the level of the file whose title is being printed. The values of INDEX in the illustrations above are examples. When a new major category, category, sub-category, or sub-sub-category is added, the appropriate "number" is entered in the INDEX field. (Note that the "index number" can contain letters and/or numbers, and that periods are used to separate numbers of the various levels of categories.)

The 10-byte INDEX field permits up to 4 levels of categories with the structure illustrated above.

Although this hierarchal structure has proved to be useful in proctice, it is presented here as an illustration of the use of the LISTS procedure, not as a model file organization.

The INDENT fields. In this example it is desireable to use indention as illustrated in the examples above. This is accomplished by using the following:

Category level	IDENT (Col. 11)	CAR_CTL (Col. 12)	Remarks
Major category	0	P	A major category is printed with no indention, at the top of a fresh page (CAR_CTL='P').
Category	1	4	A category is printed one indention from the left (i.e., beginning in print position 12+4=16) and with 3 blank lines above.
Sub-Category	2	2	A sub-category is printed with two indentions (beginnin in print position 12+2x4=20) and with 1 blank line above.
Sub-sub-category	3	1	Printed with three indentions (print position 24) and single spacing.

The Remarks in the table above explain both the INDENT and the CAR_CTL (carriage control) values.

In some cases a category may be sufficiently important to justify printing it at the tope of a new page. In such a case one uses P for the CAR_CTL character (column 12). This occurred several times in the application used for this example but is not shown in the examples.

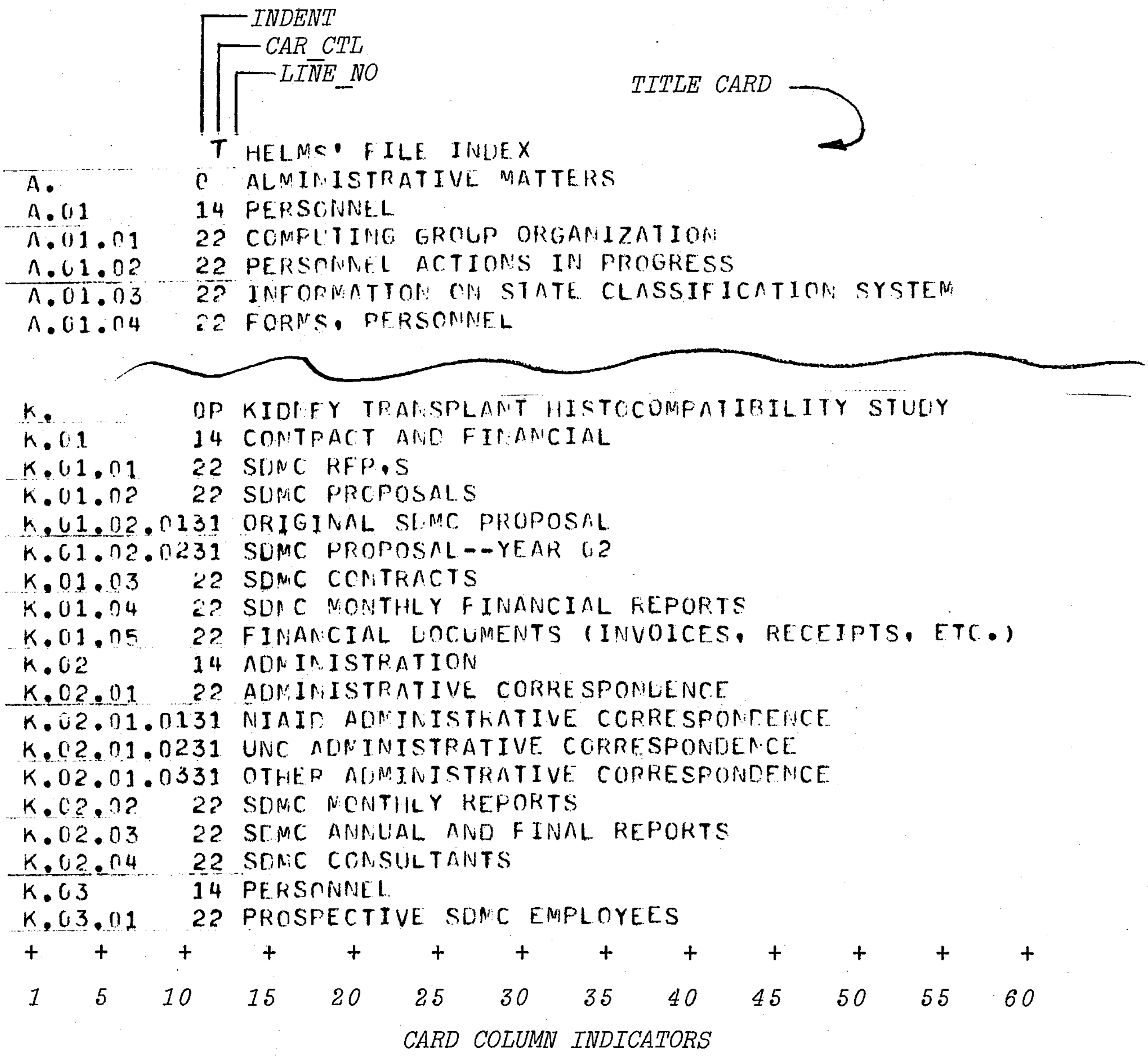
Since a title card, signaled by a T carriage control (column 12), causes the program to skip to the top of a page, a major category card following a title card should not use the P CAR_CTL character. (There would result in skipping to the top of Page 2 to print the major category line.) Instead one should use a 1 for the CAR_CTL character of a line following a title card.

The LINE-NO field. For the most part each file index title (the TEXT field) is contained in one line. In these cases the block contains only one card and the LINE-NO for that card is left blank; when the TEXT requires more than one card, subsequent cards are numbered 1, 2, ..., etc. These "continuation cards" should have the same INDENT value as the first card of the block and a CAR_CTL value of 1.

The TEXT field contains the titles of the major categories, categories, etc., as illustrated above.

Example. Figure 4 contains a listing of part of a LISTS card file used for this example. Figure 5 contains the output generated by the LISTS program from the deck shown in Figure 4.

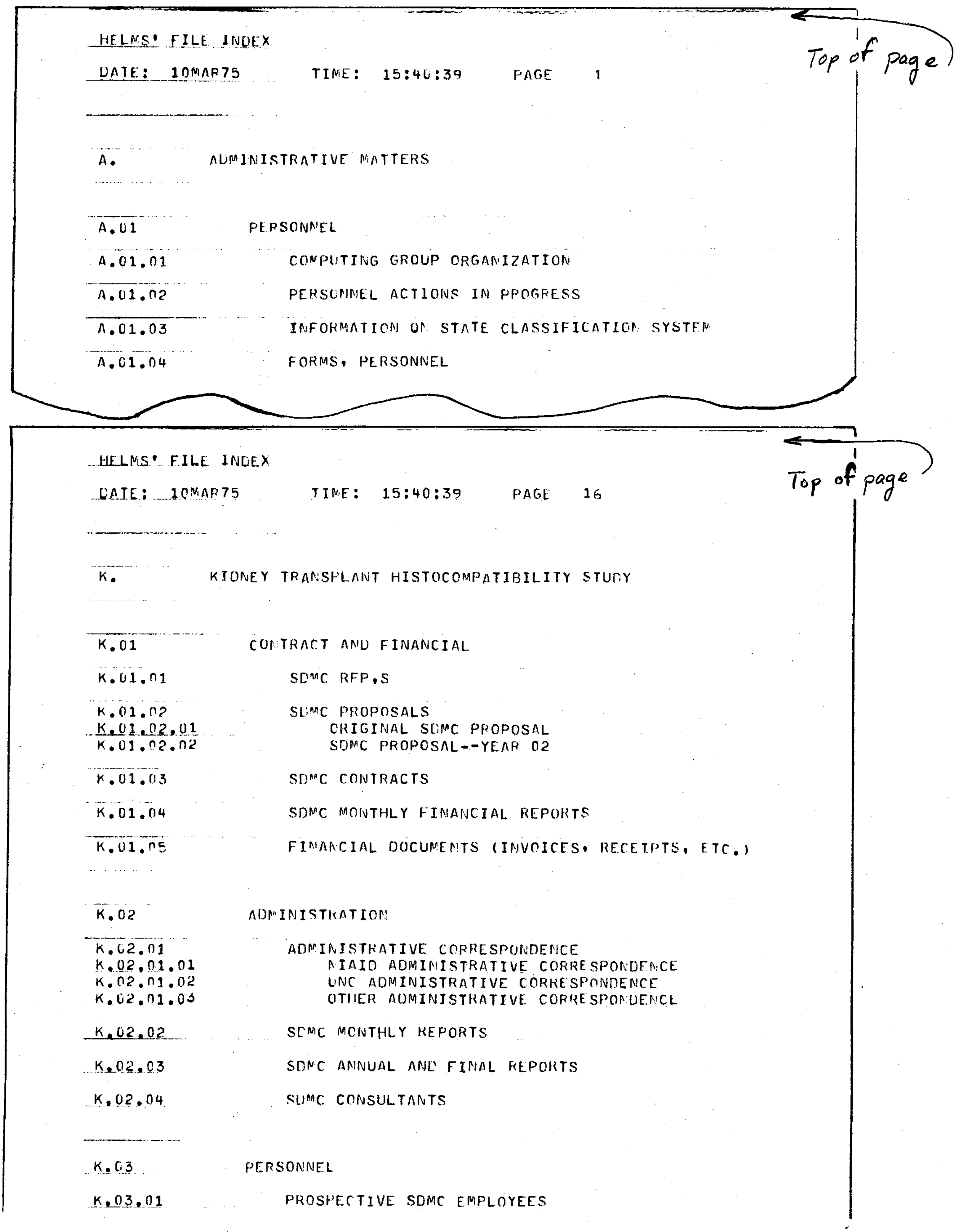
Note that the first card in Figure 4 is a title card, which is indicated by the fact that the CAR_CTL field contains a T.


INDENT
CAR_CTL
LINE_NO
TITLE CARD
+ + + + + + + + + + + + +
1 5 10 15 20 25 30 35 40 45 50 55 60
CARD COLUMN INDICATORS

```
           T HELMS' FILE INDEX
A.        0  ADMINISTRATIVE MATTERS
A.01      14 PERSONNEL
A.01.01   22 COMPUTING GROUP ORGANIZATION
A.01.02   22 PERSONNEL ACTIONS IN PROGRESS
A.01.03   22 INFORMATION ON STATE CLASSIFICATION SYSTEM
A.01.04   22 FORMS, PERSONNEL

K.        0P KIDNEY TRANSPLANT HISTOCOMPATIBILITY STUDY
K.01      14 CONTRACT AND FINANCIAL
K.01.01   22 SDMC RFP,S
K.01.02   22 SDMC PROPOSALS
K.01.02.0131 ORIGINAL SDMC PROPOSAL
K.01.02.0231 SDMC PROPOSAL--YEAR 02
K.01.03   22 SDMC CONTRACTS
K.01.04   22 SDMC MONTHLY FINANCIAL REPORTS
K.01.05   22 FINANCIAL DOCUMENTS (INVOICES, RECEIPTS, ETC.)
K.02      14 ADMINISTRATION
K.02.01   22 ADMINISTRATIVE CORRESPONDENCE
K.02.01.0131 NIAID ADMINISTRATIVE CORRESPONDENCE
K.02.01.0231 UNC ADMINISTRATIVE CORRESPONDENCE
K.02.01.0331 OTHER ADMINISTRATIVE CORRESPONDENCE
K.02.02   22 SDMC MONTHLY REPORTS
K.02.03   22 SDMC ANNUAL AND FINAL REPORTS
K.02.04   22 SDMC CONSULTANTS
K.03      14 PERSONNEL
K.03.01   22 PROSPECTIVE SDMC EMPLOYEES
```

FIGURE 4. Part of a LISTS file from the hierarchally structured file example.



```
HELMS' FILE INDEX
DATE:  10MAR75        TIME:  15:40:39     PAGE    1


A.         ADMINISTRATIVE MATTERS


A.01           PERSONNEL

A.01.01            COMPUTING GROUP ORGANIZATION

A.01.02            PERSONNEL ACTIONS IN PROGRESS

A.01.03            INFORMATION ON STATE CLASSIFICATION SYSTEM

A.01.04            FORMS, PERSONNEL
```

```
HELMS' FILE INDEX
DATE:  10MAR75        TIME:  15:40:39     PAGE   16


K.         KIDNEY TRANSPLANT HISTOCOMPATIBILITY STUDY


K.01           CONTRACT AND FINANCIAL

K.01.01            SDMC RFP,S

K.01.02            SDMC PROPOSALS
K.01.02.01             ORIGINAL SDMC PROPOSAL
K.01.02.02             SDMC PROPOSAL--YEAR 02

K.01.03            SDMC CONTRACTS

K.01.04            SDMC MONTHLY FINANCIAL REPORTS

K.01.05            FINANCIAL DOCUMENTS (INVOICES, RECEIPTS, ETC.)


K.02           ADMINISTRATION

K.02.01            ADMINISTRATIVE CORRESPONDENCE
K.02.01.01             NIAID ADMINISTRATIVE CORRESPONDENCE
K.02.01.02             UNC ADMINISTRATIVE CORRESPONDENCE
K.02.01.03             OTHER ADMINISTRATIVE CORRESPONDENCE

K.02.02            SDMC MONTHLY REPORTS

K.02.03            SDMC ANNUAL AND FINAL REPORTS

K.02.04            SDMC CONSULTANTS


K.03           PERSONNEL

K.03.01            PROSPECTIVE SDMC EMPLOYEES
```

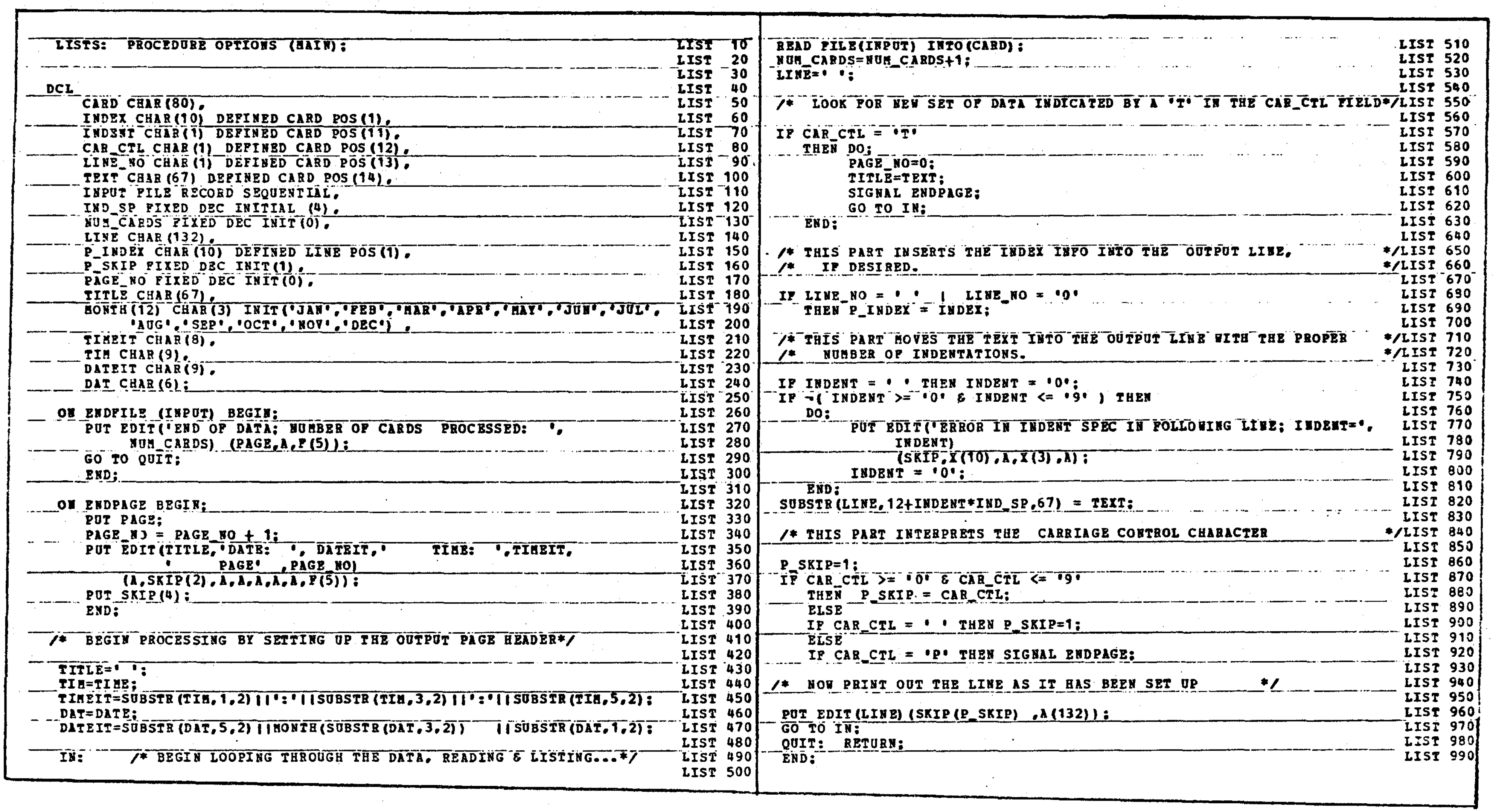
FIGURE 5. LISTS listing of the hierarchal file index example shown in Figure 4.

5. APPENDIX: TECHNICAL DETAILS

5.1. Description of Algorithm

The algorithm used by LISTS is very simple. The algorithm is outlined below. The names in all caps are variable names from the program listing, Figure 6.

1. Initialization. The time and date are manipulated into the proper format for printing page headers. The TITLE is initialized to all blanks.
2. A card image record is read into CARD, a card counter incremented, and the LINE to be printed is set to all blanks.
3. If the carriage control character, CAR_CTL (position 12 of CARD) = 'T', the TEXT of the CARD (positions 14-80) is moved to TITLE, the page-numbering is reinitialized, and a header is printed at the top of a new page by signaling ENDPAGE. Control then goes to step (2) above.
4. If the line number, LINE_NO (position 13 of CARD) is blank or zero, the INDEX field (positions 1-10 of CARD) is moved to positions 1-10 (P_INDEX) of the line to be printed (LINE).
5. The TEXT of CARD is moved to the output LINE with the proper number of indentations. A blank is INDENT is treated as a zero. If INDENT is not blank or one of the digits 1, 2, ..., 9, an error message is printed and INDENT is set to zero.
6. The carriage control character (CAR_CTL) is interpreted. (Note that CAR_CTL='T' is intercepted at step 3 and is not processed here.) If CAR_CTL='P', the printer is skipped to a new page via the SIGNAL ENDPAGE. (The ON ENDPAGE condition block actually performs skipping and page header printing.) If CAR_CTL is one of the digits 0, 1, ..., 9 the variable P_SKIP is set equal to CAR_CTL and P_SKIP lines will be skipped before the line is printed. CAR_CTL='b̸' (blank) is treated as CAR_CTL='1'. Note that CAR_CTL='0' causes the current TEXT to be printed on the same line as the previous TEXT line. Any other values of CAR_CTL are invalid and are treated as if CAR_CTL='1'.



```
 LISTS:  PROCEDURE OPTIONS (MAIN);


DCL
     CARD CHAR(80),
     INDEX CHAR(10) DEFINED CARD POS(1),
     INDENT CHAR(1) DEFINED CARD POS(11),
     CAR_CTL CHAR(1) DEFINED CARD POS(12),
     LINE_NO CHAR(1) DEFINED CARD POS(13),
     TEXT CHAR(67) DEFINED CARD POS(14),
     INPUT FILE RECORD SEQUENTIAL,
     IND_SP FIXED DEC INITIAL (4),
     NUM_CARDS FIXED DEC INIT(0),
     LINE CHAR(132),
     P_INDEX CHAR(10) DEFINED LINE POS(1),
     P_SKIP FIXED DEC INIT(1),
     PAGE_NO FIXED DEC INIT(0),
     TITLE CHAR(67),
     MONTH(12) CHAR(3) INIT('JAN','FEB','MAR','APR','MAY','JUN','JUL',
            'AUG','SEP','OCT','NOV','DEC') ,
     TIMEIT CHAR(8),
     TIM CHAR(9),
     DATEIT CHAR(9),
     DAT CHAR(6);

  ON ENDFILE (INPUT) BEGIN;
     PUT EDIT('END OF DATA; NUMBER OF CARDS  PROCESSED:  ',
          NUM_CARDS) (PAGE,A,F(5));
     GO TO QUIT;
     END;

  ON ENDPAGE BEGIN;
     PUT PAGE;
     PAGE_NO = PAGE_NO + 1;
     PUT EDIT(TITLE,'DATE:  ', DATEIT,'     TIME:  ',TIMEIT,
             '       PAGE'  ,PAGE_NO)
          (A,SKIP(2),A,A,A,A,A,F(5));
     PUT SKIP(4);
     END;

 /*  BEGIN PROCESSING BY SETTING UP THE OUTPUT PAGE HEADER*/

  TITLE=' ';
  TIM=TIME;
  TIMEIT=SUBSTR(TIM,1,2)||':'||SUBSTR(TIM,3,2)||':'||SUBSTR(TIM,5,2);
  DAT=DATE;
  DATEIT=SUBSTR(DAT,5,2)||MONTH(SUBSTR(DAT,3,2))    ||SUBSTR(DAT,1,2);

  IN:      /* BEGIN LOOPING THROUGH THE DATA, READING & LISTING...*/

 READ FILE(INPUT) INTO(CARD);
 NUM_CARDS=NUM_CARDS+1;
 LINE=' ';

 /*  LOOK FOR NEW SET OF DATA INDICATED BY A 'T' IN THE CAR_CTL FIELD*/

 IF CAR_CTL = 'T'
    THEN DO;
           PAGE_NO=0;
           TITLE=TEXT;
           SIGNAL ENDPAGE;
           GO TO IN;
       END;

 /* THIS PART INSERTS THE INDEX INFO INTO THE  OUTPUT LINE,                */
 /*   IF DESIRED.                                                          */

  IF LINE_NO = ' '  |  LINE_NO = '0'
     THEN P_INDEX = INDEX;

  /* THIS PART MOVES THE TEXT INTO THE OUTPUT LINE WITH THE PROPER        */
  /*   NUMBER OF INDENTATIONS.                                             */

  IF INDENT = ' ' THEN INDENT = '0';
  IF ¬( INDENT >= '0' & INDENT <= '9' ) THEN
     DO;
           PUT EDIT('ERROR IN INDENT SPEC IN FOLLOWING LINE; INDENT=',
                  INDENT)
                  (SKIP,X(10),A,X(3),A);
           INDENT = '0';
     END;
  SUBSTR(LINE,12+INDENT*IND_SP,67) = TEXT;

  /* THIS PART INTERPRETS THE  CARRIAGE CONTROL CHARACTER                 */

  P_SKIP=1;
  IF CAR_CTL >= '0' & CAR_CTL <= '9'
     THEN  P_SKIP = CAR_CTL;
     ELSE
     IF CAR_CTL = ' ' THEN P_SKIP=1;
     ELSE
     IF CAR_CTL = 'P' THEN SIGNAL ENDPAGE;

 /*  NOW PRINT OUT THE LINE AS IT HAS BEEN SET UP        */

  PUT EDIT(LINE) (SKIP(P_SKIP)  ,A(132));
  GO TO IN;
  QUIT:  RETURN;
  END;
```

FIGURE 6. Listing of the LISTS Program.

7. The printer is advanced P_SKIP lines and LINE is printed. Control is returned to step 2.

5.2. Definition of Fields in Input Cards

The following text defines the actions taken by the LISTS program for various values of the control fields.

INDEX (Positions 1-10). The INDEX field is printed in positions 1-10 of the output when LINE_NO= 'b̸' or LINE_NO= '0'. Otherwise positions 1-10 of the output line are blank.

CAR CTL (Position 12). If CAR_CTL= 'T', the TEXT (positions 14-80) of the card is treated as a title for page headers and the printer is skipped to the top of a new page. The page number is re-initialized to 1.

If CAR_CTL= 'n', where n is one of the digits (0, 1, ..., 9), then the line printer will be advanced n positions before the line of TEXT is printed. This leaves n-1 blank lines between the previous line and the next line. For example, if CAR_CTL= '1', the printer is advanced one line before printing, leaving no blank lines.

If CAR_CTL= '0' (zero), the next line will be overprinted on the same line as the previous line. (This can be used to underscore a line or to create a boldface effect by overprinting the same TEXT two or more times.

If CAR_CTL= 'P', the current TEXT line will be printed at the top of the next page, under the page header.

If CAR_CTL is none of the characters listed above (T, P, 0, 1, ..., 9), it is treated as if CAR_CTL= '1'. In particular, CAR_CTL= 'b̸' (blank) has the same effect as CAR_CTL= '1'.

INDENT (Position 11 of CARD). If the card is a title card ('T' in position 12) the INDENT character has no effect. Otherwise, if INDENT = 'n', where n is one of the digits (0, 1, 2, ..., 9), the TEXT is indented 4 * n + 12 positions in the printed line. The first 10 positions are reserved for the INDEX field. Positions 11-12 are blank and separate the INDEX from the TEXT. Thus, if INDENT = '0' (zero), the TEXT is printed beginning in position 12. If INDENT = '1', the TEXT is printed beginning in position 16, etc.

INDENT = 'b̸' (blank) is treated as if INDENT = '0' (zero).

All other values of INDENT (i.e., everything except 'b̸', 0, 1, ..., 9) are invalid and are treated as if INDENT = '1'. An error message is printed when an invalid value of INDENT is encountered.

LINE_NO (Position 13 of CARD). The LINE_NO field is used to sequence the cards within each block. (A block is a set of cards with identical values in the INDEX field.) The LINE_NO field can contain any valid characters. For blocks of 10 or fewer cards, the digits 0, 1, ..., 9 should be used. For blocks with more than 10 cards use the blank character ('b̸') for the LINE_NO of the first card in the block and the capital letters (A, B, C, ...) for LINE_NOs of subsequent cards.

Two LINE_NO values invoke special action. If LINE_NO= 'b̸' (blank) or LINE_NO= '0' (zero) the INDEX field is printed in positions 1-10 of the output line. For this reason the 'b̸' or '0' characters should be reserved for use as the LINE _NO of the first line (card) in each block.

TEXT (Positions 14-80 of CARD). Most of the details of how the TEXT field is manipulated are discussed above. If CAR_CTL= 'T', the TEXT field is treated as a title field and is printed at the top of each page until a new title card is processed. Otherwise, the TEXT is inserted in the output line at a position specified by INDENT and the line is printed after skipping the number of positions specified by CAR_CTL.

The TEXT field may contain any characters with valid printing symbols. The program does not process the TEXT field other than to position it in the output line and print it.