RECALL:

A Management Information Retrieval System for the Wang 2200

November 1976

U.S. Army Materiel Development and Readiness Command
HARRY DIAMOND LABORATORIES
Adelphi, Maryland 20783
The findings in this report are not to be construed as an official Department of the Army position unless so designated by other authorized documents.

Citation of manufacturers' or trade names does not constitute an official indorsement or approval of the use thereof.

Destroy this report when it is no longer needed. Do not return it to the originator.
# RECALL: A Management Information Retrieval System for the Wang 2200

**Author(s):** Howard M. Bloom

**Performing Organization Name and Address:**
Harry Diamond Laboratories
2800 Powder Mill Road
Adelphi, MD 20783

**Contract or Grant Number(s):**
HDL Project: 738671
DRCMS Code: 738017.H0.0R000

**Abstract:**
A data-base language called RECALL has been implemented on the Wang 2200 programmable desk calculator. The language was patterned after RETRIEVE, developed by Tymshare, Inc., for its time-sharing network. Differences between the two implementations are very minor. The report describes each command in the language and gives a comprehensive example to illustrate how the entire system can be used. The listing of the program allows the reader to make modifications, if necessary. The RECALL system is approved for public release; distribution unlimited.

**Keywords:** RECALL, Wang 2200, Data-base system, Programmable desk calculator, Storage and retrieval.
implemented on a 2200C calculator with 16k bytes of memory and a model 2230 disk used for temporary storage of the overlay segments needed for running the program. The system assumes that the data bases will be permanently saved on tape cassette.
<table>
<thead>
<tr>
<th>CONTENTS</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>1. INTRODUCTION</td>
<td>5</td>
</tr>
<tr>
<td>2. DATA-BASE DESCRIPTION</td>
<td>5</td>
</tr>
<tr>
<td>2.1 Command Description</td>
<td>6</td>
</tr>
<tr>
<td>2.2 Expressions and Conditions</td>
<td>9</td>
</tr>
<tr>
<td>2.3 Line Continuation</td>
<td>11</td>
</tr>
<tr>
<td>2.4 Field and Expression Lists</td>
<td>11</td>
</tr>
<tr>
<td>2.5 Running RECALL System</td>
<td>11</td>
</tr>
<tr>
<td>2.6 Hard-Copy Capability</td>
<td>12</td>
</tr>
<tr>
<td>2.7 Data-Base Maintenance</td>
<td>12</td>
</tr>
<tr>
<td>2.8 Tape Handling</td>
<td>12</td>
</tr>
<tr>
<td>2.9 Difference between Tymshare RETRIEVE and Wang RECALL</td>
<td>12</td>
</tr>
<tr>
<td>3. DATA-BASE CREATION AND ACCESS COMMANDS</td>
<td>14</td>
</tr>
<tr>
<td>3.1 Creating Data Base</td>
<td>14</td>
</tr>
<tr>
<td>3.2 Adding to Data Base</td>
<td>15</td>
</tr>
<tr>
<td>3.3 Loading Data Base</td>
<td>16</td>
</tr>
<tr>
<td>3.4 Merging Data Bases</td>
<td>16</td>
</tr>
<tr>
<td>3.5 Saving Data Bases</td>
<td>16</td>
</tr>
<tr>
<td>3.6 Displaying Data-Base Structure</td>
<td>17</td>
</tr>
<tr>
<td>3.7 Displaying Data-Base Size</td>
<td>17</td>
</tr>
<tr>
<td>4. DATA-BASE DISPLAY COMMANDS</td>
<td>17</td>
</tr>
<tr>
<td>4.1 Displaying Set of Records</td>
<td>17</td>
</tr>
<tr>
<td>4.2 Counting Records</td>
<td>18</td>
</tr>
<tr>
<td>4.3 Summing Expressions</td>
<td>18</td>
</tr>
<tr>
<td>4.4 Averaging Expressions</td>
<td>18</td>
</tr>
<tr>
<td>5. DATA-BASE UPDATING COMMANDS</td>
<td>18</td>
</tr>
<tr>
<td>5.1 Changing Record Fields</td>
<td>18</td>
</tr>
<tr>
<td>5.2 Deleting Records</td>
<td>19</td>
</tr>
<tr>
<td>5.3 Replacing Records</td>
<td>19</td>
</tr>
<tr>
<td>5.4 Sorting Records</td>
<td>19</td>
</tr>
<tr>
<td>6. REPORT GENERATION</td>
<td>20</td>
</tr>
<tr>
<td>7. EXAMPLE OF RECALL SYSTEM</td>
<td>22</td>
</tr>
</tbody>
</table>
APPENDICES

A. --- Using RECALL Data Base in Basic Program .................................. 33
B. --- Listing of Source Code ................................................................. 37

DISTRIBUTION .......................................................................................... 57

FIGURES

1  Example of RECALL run for CREATE and LIST commands .................. 23
2  Example of RECALL run for CHANGE and REPLACE commands ......... 24
3  Example of RECALL run for SUM and APPEND commands ............... 25
4  Example of RECALL run for SORT and MERGE commands .............. 26
5  Example of RECALL run for SIZE, DELETE, and TAPE APPEND commands ...................................................................................... 27
6  Example of RECALL run for STRUCTURE, LOAD, COUNT, and AVERAGE commands .............................................................................. 28
7  Example of REPORT command ............................................................... 29
8  Example of REPORT command continued .......................................... 30
9  Example of REPORT command to generate tape .................................. 31

TABLES

I  Data Base Creation and Access Commands ........................................ 7
II Display Commands .................................................................................. 7
III Updating Commands ............................................................................. 8
1. INTRODUCTION

A data-base language (DBL) called RECALL has been implemented on the Wang 2200 programmable desk calculator. The language was patterned after RETRIEVE, originally designed for the Tymshare computer system. Because of the ease in using the language, it was believed that RETRIEVE would be an ideal DBL to use with the Wang system. Hence, the language has been implemented in its entirety with just a few small changes. This report only briefly describes the language itself. Most of the emphasis is placed on how to run the system on the Wang 2200 and what the differences are between the original Tymshare version and the new Wang version. The RETRIEVE\textsuperscript{1} manual details the language.

The Wang RECALL system has been implemented on a 2200C calculator with 16k bytes of memory. The system uses temporary storage on the last 320 sectors of both the fixed and the removable disk platters on the model 2230 disk. The system's main program requires approximately 12k bytes for the common subroutines and variables. Four 3k-byte segments are overlaid when necessary for the implementation of the various commands. The various segments are stored on the fixed disk platter. The system assumes that the data are saved on tape cassette and is modelled around this concept.

A technique for using RECALL data bases in the BASIC program is described in appendix A. The source code for RECALL is listed in appendix B.

2. DATA-BASE DESCRIPTION

The RECALL system is designed to store a given collection of related data and to access and update this information, the "data base." A data base is divided into "records"; each record consists of one or more "fields"; each field contains a datum such as a number or a string of characters. All records in a data base contain the same number of fields arranged in the same order.

A field may contain data from 1 to 32 characters in length, and a record may contain up to 120 characters. There can be a maximum of 15 fields in a record. The entire data base can contain approximately 78,000 characters, i.e., the amount of storage on a Wang 1251 tape cassette. The data-base name can be one to eight characters in length.

\textsuperscript{1}RETRIEVE, Tymshare, Inc. Palo Alto, CA (1971).
A field is referenced by its name, which can be one to eight characters. It must begin with a letter and can contain only letters, digits, and periods. The following names cannot be used because they are reserved for commands: AND, FOR, FROM, IN, NOT, ON, OR, and WITH.

A field can be of either character or numeric type. A character field can contain any character on the keyboard, whereas a numeric field can contain only numbers. A character field is specified by the letter C when defining the data base. There are two kinds of numeric fields, integer and noninteger. An integer field cannot contain decimal points, whereas a noninteger field can. However, a noninteger field cannot use the E format number (allowable in Tymshare). An integer field is specified by the letter I; a noninteger field, N.

2.1 Command Description

The RECALL system is operated by the specifying of a set of commands (tables I to IV) to manipulate the data base. The general format of a command is the following:

```
[range list]    command    [FOR condition]
```

Only the first three characters need be typed for all commands except REPORT (REPO must be used). The range list allows the user to specify which records within the data base should be considered for that command. The list can be of three types:

a. A single record number (e.g., "45")

b. A range of records, consisting of a pair of numbers separated by a colon (e.g., "15:20" means records 15 to 20)

c. Any combination of single number or range; the record numbers are separated by semicolons (e.g., "15;18:20;25" means records 15, 18, 19, 20, 25)

Up to five combinations (four semicolons) can be specified. If a range list is not specified for a given command, then it is assumed that the whole data base is to be used.

The Wang implementation of RECALL uses a semicolon rather than a comma as a separator. Warning: A comma should never be typed for any command within the RECALL system. The blank space is not acceptable as a separator for the range list. The "§" cannot be used to specify the last record.
# TABLE I. DATA BASE CREATION AND ACCESS COMMANDS

<table>
<thead>
<tr>
<th>Command</th>
<th>Function</th>
</tr>
</thead>
<tbody>
<tr>
<td>CREATE data base</td>
<td>Creates new data base with given name (destroys presently active data base in system)</td>
</tr>
<tr>
<td>APPEND (FROM data base)</td>
<td>Allows records to be added to data base from keyboard or from data base on tape</td>
</tr>
<tr>
<td>LOAD data base</td>
<td>Loads data base from tape into storage as active data base</td>
</tr>
<tr>
<td>MERGE ON field list (FROM data base)</td>
<td>Merges data base on tape with present data base</td>
</tr>
<tr>
<td>[range] SAVE TO data base [FOR option]</td>
<td>Saves present data base on tape</td>
</tr>
<tr>
<td>STRUCTURE</td>
<td>Displays field structure of data-base records</td>
</tr>
<tr>
<td>SIZE</td>
<td>Displays number of records in data base</td>
</tr>
<tr>
<td>QUIT</td>
<td>Ends RECALL program</td>
</tr>
</tbody>
</table>

# TABLE II. DISPLAY COMMANDS

<table>
<thead>
<tr>
<th>Command</th>
<th>Function</th>
</tr>
</thead>
<tbody>
<tr>
<td>[range] LIST [field list] [FOR option]</td>
<td>Displays data-base records for specified fields with record numbers and field headings included</td>
</tr>
<tr>
<td>[range] PRINT [field list] [FOR option]</td>
<td>Same as LIST, except no record numbers</td>
</tr>
<tr>
<td>[range] FAST [field list] [FOR option]</td>
<td>Same as PRINT, except no heading</td>
</tr>
<tr>
<td>[range] COUNT [FOR option]</td>
<td>Counts number of records</td>
</tr>
<tr>
<td>[range] SUM expression list [FOR option]</td>
<td>Displays data base totals for each expression (five max)</td>
</tr>
<tr>
<td>[range] AVERAGE expression list [FOR option]</td>
<td>Displays data-base average values for each expression (five max)</td>
</tr>
</tbody>
</table>
### TABLE III. UPDATING COMMANDS

<table>
<thead>
<tr>
<th>Command</th>
<th>Function</th>
</tr>
</thead>
<tbody>
<tr>
<td>[range] CHANGE [field list]</td>
<td>Allows selected fields to be changed in record</td>
</tr>
<tr>
<td>[FOR option]</td>
<td></td>
</tr>
<tr>
<td>[range] DELETE [FOR option]</td>
<td>Deletes selected records</td>
</tr>
<tr>
<td>[range] REPLACE field₁ with</td>
<td>Replaces selected fields with any</td>
</tr>
<tr>
<td>expression₁ ; field₂ WITH</td>
<td>desired expression</td>
</tr>
<tr>
<td>expression₂ ; . . . [FOR</td>
<td></td>
</tr>
<tr>
<td>option]</td>
<td></td>
</tr>
<tr>
<td>SORT ON field list</td>
<td>Ascending sort of data base</td>
</tr>
</tbody>
</table>

### TABLE IV. REPORT GENERATION COMMANDS—-[RANGE LIST] REPORT ["FOR" CONDITION]

<table>
<thead>
<tr>
<th>RECALL prompt</th>
<th>User responses allowed</th>
</tr>
</thead>
<tbody>
<tr>
<td>1 REPORT OUTPUT TO</td>
<td>T (to display) or data base name (to save on tape)</td>
</tr>
<tr>
<td>2 REPORT FORM NAME</td>
<td>Name (if blank, skip to 4)</td>
</tr>
<tr>
<td>3 UPDATE REPORT FORM</td>
<td>Y or N (if N, old report form is used)</td>
</tr>
<tr>
<td>4 HEADING</td>
<td>Y or N</td>
</tr>
<tr>
<td>5 DOUBLE SPACE</td>
<td>Y or N</td>
</tr>
<tr>
<td>6 TOTALS</td>
<td>Y or N (if N, skip to 10)</td>
</tr>
<tr>
<td>7 SUBTOTALS</td>
<td>Y or N (if N, skip to 9)</td>
</tr>
<tr>
<td>8 BY ITEMS</td>
<td>List of items (record fields) on which to subtotal (three fields max)</td>
</tr>
<tr>
<td>9 SUMMARY REPORT ONLY</td>
<td>Y or N</td>
</tr>
<tr>
<td>10 COL WIDTH:CONTENTS</td>
<td>Column width; expression (requests description of each column in report—end with RETURN to column prompt (15 columns max)</td>
</tr>
<tr>
<td>11 COL HEADING</td>
<td>Heading (requests column headings) (10 characters per row max, three rows max)</td>
</tr>
<tr>
<td>12 COLUMNS-TOTAL; NO OF DECIMAL PLACES</td>
<td>Y or N; number of decimal places (prompted for each numeric column) (five columns max)</td>
</tr>
<tr>
<td>13 COLUMNS NO OF DECIMAL PLACES</td>
<td>Number of decimal places (asked if totals were not requested)</td>
</tr>
</tbody>
</table>
2.2 Expressions and Conditions

The FOR condition allows the user to further specify on which records a command is to operate. For example, LIST FOR SIZE > 100 lists all records in the data base for which the field name SIZE has a value greater than 100.

A condition is a combination of expressions. Hence, first expression must be defined.

A numeric expression can be any of the following:

a. A number
b. A numeric field name
c. Any meaningful combination of number and numeric field name, using arithmetic operators

The following operators are allowed: +, *, /, +, and -. Warning: Unary negation is not allowed. One must use "0-5" to represent "-5."

A character expression can be any of the following:

a. A string enclosed in single quotation marks (Double quotation marks are not allowed.)
b. Any character field name
c. Any meaningful combination of string and field name, with the operator + used for string concatenation

A condition may consist of numeric expressions or character expressions related by these relational operators: < (less than), > (greater than), = (equal to), # (unequal to), <= (less than or equal to), >= (greater than or equal to).

The long form (e.g., LESS THAN for <) allowable in Tymshare RETRIEVE was not implemented in Wang RECALL.

Another condition consists of character expressions related by the operator IN (or NOT IN). This operator specifies that one string is (or is not) contained in another.

The relational expressions can be combined by using the following logical words, listed in descending order of precedence:
NOT A is true if A is false.

AND A AND B is true only if A and B are both true.

OR A OR B is true if either A or B or both are true.

Conditions in RECALL are always preceded by the FOR modifier and always have a value of true or false.

Parentheses may be used in expressions and conditions to specify the order of operations. For example, "(4+3)*5" has the value 35, but "4+3*5" has the value 19.

In evaluating an expression or condition, three rules govern the order in which operations are performed:

a. All operations with the innermost set of parentheses are performed first; then those within each succeeding outer set are performed.

b. Without violating rule a, operations are performed in the following order:

+ * and /
+ and -
<, <=, >, >=, =, 
IN, NOT IN

NOT

AND

OR

c. Without violating rules a and b, operations are performed left to right.
2.3 Line Continuation

If there is not enough room to enter the entire command (or input data record) on one line, the user can type the character "&" and then proceed to the next line. If this character is not used and the command runs over into the next line, the remaining portion of the command is ignored. A field, name, number, or string cannot be continued from one line to the next. The user should backspace to erase the information just entered, then type the "&," and begin the information on the next line.

2.4 Field and Expression Lists

The field list contains one or more field names separated by semicolons. (In Tymshare, commas are used.) The expression list contains one or more expressions separated by semicolons.

2.5 Running RECALL System

The user should check that the equipment is turned on, including the master switch for the central processing unit (CPU) and cathode ray tube (CRT). Someone authorized must turn on the disk. If he desires hard copy, the user should turn on the line printer (depress the POWER and SELECT buttons).

The system is loaded by keying

```
LOAD DCF "RECALL"
```

```
RUN
```

To end every command, the RETURN button is keyed.

The system responds with this message:

```
RECALL SYSTEM-REVISION 8/22/75
```

```
DATE (MM/DD/YY)?
```

The first line gives the present revision date of the system, and the second line requests the present date. This date subsequently is used if a report is generated or a data base is saved on tape. The user then types in the date, using the digit code for the month, day, and year.
The system is now ready for the user to manipulate the database by typing commands. The system responds "COMMAND?" and the user inputs the desired command. The system performs the appropriate operations and then prompts for a new command. If the format of the command is wrong, the system prints "ILLEGAL COMMAND" and the user must reenter his command. When the user is finished with the system, he should enter the command QUIT. This command halts the system.

2.6 Hard-Copy Capability

The system is designed to generate all displays at the CRT. If the user wishes to print out on a hard-copy unit, he should respond to the command prompt by keying HALT/STEP and then typing in the desired command. The system responds with a colon. The user can type

SELECT PRINT XXX(YYY)

to indicate the hard-copy unit XXX (e.g., "215") and the column width YYY (e.g., "132"). He then keys "CONTINUE." The system generates the printout at the hard-copy unit. When he wants to return the display to the CRT, the user can again respond to the command prompt by keying HALT/STEP and then the command. After that, he should type this:

SELECT PRINT 005(64)

2.7 Data-Base Maintenance

The system always operates on the active data base that is stored in the memory and on a disk. Whenever any operation is performed to alter the data base (e.g., sorting or deleting), the active data base is changed and the previous state is destroyed. Warning: The user should save pertinent copies of the data base on tape before commanding a status change. In Tymshare, the old version of the data base is saved under a different name, and the user is asked if he wishes to maintain this old version.

2.8 Tape Handling

For all commands requiring tapes, the system prints the message:

MOUNT TAPE AND TYPE CONTINUE

2.9 Differences between Tymshare RETRIEVE and Wang RECALL

The Tymshare (T) RETRIEVE and Wang (W) RECALL differ as follows:
a. T uses a comma or blank as a field separator; W, a semicolon.

b. T uses $ for the last record indicator; W does not.

c. In T, names have 10 characters; fields, 24; and records, 185; there can be 18 fields. In W, names can be only 8 characters; fields 32; and records, 132. There can be only 15 fields.

d. A floating point numerical format (E) is allowed in T, but not in W.

e. The ALL, BINARY, IF, RECNO, RESULTS, SCRAMBLED, SEQUENCED, FIELDS, and SYMBOLIC key words have been implemented in T, but not in W.

f. The commands SORT and MERGE use BY in T, but ON in W.

g. The command BASE is used in T, but only LOAD is used in W.

h. SORT and MERGE lists need not be specified and can be greater than three fields in T; SORT and MERGE lists must be specified and cannot be greater than three fields in W.

i. The operator NOT can be used with IN and OUT in T, but only with IN in W.

j. The long form for relationships in T is not used in W.

k. Lines are continued by using the line continuation key in T, but the operator "&" in W.

l. In T, 10 columns max can be totalled in the command REPORT, and 10 fields max can be used to determine subtotals. In W, five columns max can be totalled in the command REPORT, and three fields max can be used to determine subtotals.

m. In T, 10 fields max and, in W, 5 fields max can be summed or averaged by using the command SUM or AVERAGE.

n. In FOR conditions, string constants can use double quotation marks in T, but only single quotation marks in W.

o. Editing is done with control characters in T, but with the BACKSPACE and LINE/ERASE keys in W.

p. An old version of a data base saved in T is not saved in W when a command such as SORT is executed.
q. The command LOAD can be used to create a new data base in T, but not in W.

r. The space between fields generated in the command REPORT when saving the report in a file is included in the field width in T, but not in W.

s. The expression list used in the commands SUM and AVERAGE need not be specified in T, but must be in W.

t. The command MODIFY is implemented in T. The command CHANGE must be used in W.

u. In T, the old value is retained when CHANGE is commanded. In W, each field specified for an edit in the command CHANGE must be specifically entered when prompted; if not, a blank field is stored.

v. The headings on a report can have 20 characters max for any column on any line in T, but 10 in W.

3. DATA-BASE CREATION AND ACCESS COMMANDS

Certain commands create and access the data base.

3.1 Creating Data Base

The command CREATE [data base] creates the data base. If the user does not specify the data-base name (one to eight characters), the system prompts with "DATA BASE?" RECALL does not allow either binary or scrambled data bases as RETRIEVE does.

As soon as the command is entered, any data base active in the system is destroyed. The system prompts with

PLEASE TYPE IN THE STRUCTURE OF YOUR DATA BASE

FIELD NAME;WIDTH;TYPE;DECPL

The system then prompts with a field number, and the user enters the desired name; width; type (I, N, or C); and, if N, the number of decimal places. If the user prematurely ends the line without entering all the information or enters incorrect information, the system prompts for each piece of information separately until the entire field is entered. The user responds to the FIELD prompt by keying the command RETURN when all the fields have been entered.
To prompt for the actual record data, the system first displays the sequence of field names. The user enters each record that ends with a command RETURN. He ends the record prompt by keying RETURN at the beginning of a record. (The character / symbolizes RETURN in this report.) If the user prematurely ends the record without entering all the fields or if he enters an incorrect field value, the system prompts for the remaining field values in the record by first displaying the field name. The record can be continued on the next line by using the operator "&." The system prints the number of records stored for the data base, as in this example:

```
1  CUSTOMER;15;C /
2  AGE;3;I /
3  PRICE;8;N;2 /
4  
CUSTOMER;AGE;PRICE

APPLE;20;18.10 /
PEAR;14;22 /
GRAPE;84;171.45 /
/
3 RECORDS
```

The values of all character fields are stored left justified in the data base. The values of the numeric fields are stored right justified. If a numeric field is specified as having three decimal places, then, when the value is entered, the appropriate number of trailing zeros is added (with a possible decimal point) before the value is stored in the data base.

The data base is now defined and can be manipulated by all the other commands. However, this active data base does not have any permanent storage. Only the command SAVE allows a data base to be saved.

### 3.2 Adding to Data Base

The command APPEND (FROM data base) allows the user to add records to the present data base. If the "FROM [data base]" is excluded, the system prompts for the records to be entered from the keyboard as in the command CREATE. If a data-base name is included, the system prompts with this:
The user loads the tape, and the system reads all the records from the tape and adds them to the active data base. The records must be in the exact format as the active data base. If the name on the tape disagrees with the name in the command, the system prints "WRONG DATA BASE" and indicates an illegal command. After all the records are loaded, the system prints the new record size of the data base. The RETRIEVE modifiers SEQUENCED and FIELDS are not implemented in RECALL.

3.3 Loading Data Base

The command LOAD [data base] is used to load a data base that has previously been saved on tape. If the name on the tape disagrees with the name on the command, the system prints "WRONG DATA BASE" and indicates an illegal command. The system displays "DATABASE [name] LAST SAVED [date] HAS BEEN LOADED."

If the user does not type the data-base name, the system prompts with "DATABASE?"

This command differs slightly from that in RETRIEVE: in RECALL, the command can be used only to load a previously created data base, not to create a new data base (also, the word BASE cannot be used as a substitute for LOAD).

3.4 Merging Data Bases

The command MERGE ON [field list] FROM [data base] allows the user to merge data from a data base stored on tape into the active data base. The two data bases must be presorted in the order that is desirable for the merge. The two data bases also must have identical record structures. The field list can contain from one to three fields in the order of the desired merge, as in this example:

MERGE ON AGE, SALARY FROM PERSON

(In RETRIEVE, the ON field list is optional.) In the example, the data base PERSON is merged with the active base with respect first to AGE and then to SALARY.

3.5 Saving Data Bases

The command SAVE TO [data base] saves the active data base on tape. It is one of the commands that can use only a selected set of records. The system prompts with
and then saves the selected set of records.

3.6 Displaying Data-Base Structure

The command STRUCTURE is used to display the field description of the records. The heading

FIELD; TYPE; WIDTH; NAME

is displayed, and then the field information is given. If the field is numeric (N), the width includes the number of decimal places.

3.7 Displaying Data-Base Size

The command SIZE displays the number of records in the database.

4. DATA-BASE DISPLAY COMMANDS

Commands are available for displaying selected information from the database. All commands can have a record range list and FOR condition.

4.1 Displaying Set of Records

The commands LIST, PRINT, and FAST are all used to display a set of records. The command LIST prints out the field names as a heading and includes the record numbers. The command PRINT does not have record numbers, and the command FAST does not have record numbers or a heading. If no field list is given, all fields for each record are displayed in the order that they are stored. The field width is used to determine the space used in displaying each field. One extra space is placed between each field for clarity. If a field list is given, the display contains the data values for each record in the order that they are specified in the field list. The number of records displayed is given after the records are displayed.

For example, if the data base contains the fields CUSTOMER, AGE, and PRICE and record 5 contains CITRUS; 45; 21.23, the command 5FAST produces this:

CITRUS  45    21.23

The command 5FAST PRICE; AGE produces this:

21.23    45
4.2 Counting Records

The command COUNT is used to determine the number of records that satisfies a given condition. The system displays the record count. Without the FOR condition, the command COUNT is identical to the command SIZE. This is an example:

COUNT FOR SALARY>4500 AND AGE<29

4.3 Summing Expressions

The command SUM expression list is used for totalling specified numeric fields or expressions. There can be five expressions max in the list. For example,

SUM SALARY;AGE*SALARY FOR AGE>30

computes the sum for all salaries and the sum of the product of the age times the salary for all records where the age is greater than 30.

The system prints out the expression list before displaying the sums. The number of records used also is displayed. Unlike in RETRIEVE, the expression list must be specified and only numeric fields can be used in RECALL.

4.4 Averaging Expressions

The command AVERAGE expression list is used for averaging expressions. The usage is identical to that of the command SUM.

5. DATA-BASE UPDATING COMMANDS

Commands update the information in the data base.

5.1 Changing Record Fields

The command CHANGE [field list] allows the user to change selected records or selected fields within a record. If the field list is not specified, the entire record is updated. If the list is given, only those fields are updated. For the range of specified records, the system displays the following:

[field name] [old value]?
The user enters the new value. This prompt is repeated for each specified field in the record. As an example, assume that the record for the customer APPLE contains 32 as his age, and the value should be updated to 35.

```
CHANGE AGE FOR CUSTOMER = 'APPLE'
```

The system responds with the following line up to the colon:

```
AGE     32? 35
```

The user responds with "35," and the record is updated. The command MODIFY (the same as the command CHANGE, but without the old value displayed) is not implemented.

### 5.2 Deleting Records

The command DELETE deletes a selected set of records based on the range list and FOR condition. If both options are omitted, the entire database is deleted.

### 5.3 Replacing Records

The command REPLACE \([field_1 \text{ WITH expression}_1] [,field_2 \text{ WITH expression}_2, \ldots]\) allows the user to replace selected fields with any desired expression. The command is especially useful when the user wishes to change a number of records in the same way, since the user need not enter the changes for each record separately as for the command CHANGE. For example,

```
REPLACE AGE WITH AGE*2
```

doubles the value of all ages in the database. AGE is immediately updated, so that if it appears in a later expression in the command REPLACE, the new value is used.

### 5.4 Sorting Records

The command SORT ON [field list] allows the user to perform an ascending sort on as many as three fields in a database. The field list must be included. (It may be omitted in RETRIEVE.)
6. REPORT GENERATION

The command [range list] REPORT [FOR condition] initiates report generation. If neither the range list nor FOR condition option is used, all the records in the data base are reported.

After the command is issued, the system dialogues with the user (table IV) after first requesting the user to mount a report tape.

a. If the user responds T, the report appears on the line printer. Otherwise, the report is saved on tape as a data base with the specified name. All the records specified in the command are saved using the fields defined in prompt (table IV). This tape can then be used to set up a new data base. The user creates the new data base with the command CREATE and then uses the command APPEND to read the records from the tape. This feature is especially helpful in restructuring a data base.

b. The name specified by the user is shown on all report forms.

c. If the answer is NO, the system gathers all the necessary information from the report tape.

d. If the answer is YES, the system contains column headings and is formatted into pages 8-1/2 x 11 in. (21.6 x 27.9 cm).

e. If the answer is YES, only the body of the report is double spaced; headings and totals are single spaced.

f. If the answer is YES, the system accumulates totals of report columns containing numeric data. The user specifies which columns of the report are to be totalled by responding to the COLUMNS TOTALS question. Five columns max can be totalled.

g. If the answer is YES, the report shows subtotals for numeric report columns each time that the value of a selected data-base field changes.

h. A list of fields (separated by semicolons) is given for which, after a change in value for any field, a subtotal is generated. Three fields max can be specified.

i. If the answer is YES, only total and subtotal headings and values are shown on the report. Individual records do not appear.

j. The column contents are specified. The user ends the prompt with the command RETURN following the column number, as in this example:
Fifteen columns max can be specified. The system displays a blank space between each column. Unlike in RETRIEVE, these extra columns do not count when a data-base tape is generated.

k. The column headings are specified. The system prompts with the column number. By being separated with slashes, the heading can be spread over three lines. Each line can be 10 characters max, as in this example:

```
COL HEADING
  1 LOT/IDENT
  2 COST/
```

The actual page heading looks like this:

```
LOT  COST  IDENT
```

1. The columns' totals and decimal place information are requested for all the numeric fields. The system prompts with the appropriate column number, as in this example:

```
COLUMNS-TOTALS; NO SPACE OF DECIMAL PLACES
  2 Y;2/
```

The number of decimal places must be specified. If it is specified as zero, the field is considered integer. In this example, column 1 is not prompted. This field is character and not numeric.

m. This question is asked only if the user has not requested totals. During the dialogue, if the display output feature is chosen, the program is stopped, and the user is requested to type the select print option desired.

If the heading has been requested, the following block of information appears in the upper left corner of every page.
If subtotals have been requested, the following appears before each new subtotal group:

* [field name]: [field value]

The "*" can also be either "**" or "***," depending on the line of the subtotal.

This follows the subtotal group:

*TOTAL FOR [field name]: [field value]

The set of totals appears under the appropriate columns. If totals have been requested, the following appears at the end of the report:

** GRAND TOTAL **

The set of totals appears under the appropriate columns.

7. EXAMPLE OF RECALL SYSTEM

Figures 1 to 9 illustrate the various capabilities of RECALL.
LOAD DCR "RECALL"

RUN

RECALL SYSTEM - REVISION 8/22/75
DATE (MM/DD/YY)? 8/22/75

COMMAND? CREATE FUZEFIE
PLEASE TYPE IN STRUCTURE OF DATA BASE

NAME; WIDTH; TYPE; DECPL
1  MFR: 5; C
2  LOT: 10; C
3  SIZE: 4; I
4  REJECTS; 7; I
5  REJ. SIZE: 8; N; 2
6  

MFR   ; LOT   ; SIZE   ; REJECTS ; REJ. SIZE
? RVB; R7001; 100; 2; 0
? HP; HP001; 105; 3; 0
? UCR; UCR3001; 80; 4; 0
? HAM; H1001; 200; 5; 0
? RVB; R7002; 102; 3; 0
? RVB; R7003; 92; 1; 0
? HP; HP002; 103; 2; 0
? UCR; UCR3002; 81; 3; 0

8 RECORDS

COMMAND? LIST

RECN MFR   LOT   SIZE   REJECTS   REJ. SIZE
   1 RVB   R7001   100     2        0.00
   2 HP    HP001   105     3        0.00
   3 UCR   UCR3001 80      4        0.00
   4 HAM   H1001   200     8        0.00
   5 RVB   R7002   103     3        0.00
   6 RVB   R7003   92      1        0.00
   7 HP    HP002   103     2        0.00
   8 UCR   UCR3002 81      3        0.00

8 RECORDS

COMMAND? PRINT MFR; SIZE

MFR   SIZE
RVB   100
HP    105
UCR   80
HAM   200
RVB   103
RVB   92
HP    103
UCR   81

8 RECORDS

Figure 1. Example of RECALL run for CREATE and LIST commands.
COMMAND? 1:3:5 LIST MFR

RECN MFR

1 RYB
3 UCR
4 HAM
5 RYB

4 RECORDS
COMMAND? 3:4 CHANGE REJECTS; SIZE FOR MFR=’UCR’
REJECTS 4? 5
SIZE 80? 84
COMMAND? PRINT LOT; REJECTS FOR REJECTS>3

LOT    REJECTS

UCR3001  5
H1001    8

2 RECORDS
COMMAND? DELETE FOR MFR=’HAM’
COMMAND? REPLACE SIZE* WITH 2*SIZE; REJ. SIZE &
? WITH <REJECTS/SIZE>*100 FOR REJECTS>0
COMMAND? LIST

RECN MFR   LOT   SIZE    REJECTS     REJ. SIZE

1 RYB   R7001  200    2      1.00
2 HP     HP001 210    3      1.44
3 UCR   UCR3001 168    5      2.99
4 RYB   R7002  206    3      1.44
5 RYB   R7003  184    1      55
6 HP     HP002 206    2      .99
7 UCR   UCR3002 162    3      1.88

7 RECORDS

Figure 2. Example of RECALL run for CHANGE and REPLACE commands.
COMMAND? SUM REJECTS FOR MFR='RYB'
  REJECTS FOR MFR='RYB'
  6;
  3 RECORDS
COMMAND? SUM REJECTS*SIZE
  REJECTS*SIZE
  3570;
  7 RECORDS
COMMAND? APPEND
MFR  ;LOT  ;SIZE  ;REJECTS  ;REJ. SIZE;
? HP; HP003; 106; 5; 0
? RYB; R7004; 95; 2; 0
? UCR
LOT  ? UCR3003
SIZE  ? 90
REJECTS ? 6
REJ. SIZE? 0
  10 RECORDS
COMMAND? REPLACE REJ. SIZE WITH (REJECTS/SIZE)*100 &
  FOR REJ. SIZE=0
COMMAND? LIST
RECON MFR  LOT  SIZE  REJECTS  REJ. SIZE
  1 RYB  R7001  200  2   1.00
  2 HP    HP001  210  3   1.44
  3 UCR  UCR3001 168  5   2.99
  4 RYB  R7002  206  3   1.44
  5 RYB  R7003  184  1   .55
  6 HP    HP002  206  2   .99
  7 UCR  UCR3002 162  3   1.88
  8 HP    HP003  106  5   4.77
  9 RYB  R7004  95   2   2.11
 10 UCR  UCR3003 90   6   6.66
  10 RECORDS

Figure 3. Example of RECALL run for SUM and APPEND commands.
```plaintext
COMMHND? SORT ON MFR; LOT
COMMHND? LIST

<table>
<thead>
<tr>
<th>RECNO</th>
<th>MFR</th>
<th>LOT</th>
<th>SIZE</th>
<th>REJECTS</th>
<th>REJ SIZE</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>HP</td>
<td>HP001</td>
<td>210</td>
<td>3</td>
<td>1.44</td>
</tr>
<tr>
<td>2</td>
<td>HP</td>
<td>HP002</td>
<td>206</td>
<td>2</td>
<td>.99</td>
</tr>
<tr>
<td>3</td>
<td>HP</td>
<td>HP003</td>
<td>106</td>
<td>5</td>
<td>4.77</td>
</tr>
<tr>
<td>4</td>
<td>RYB</td>
<td>R7001</td>
<td>200</td>
<td>2</td>
<td>1.00</td>
</tr>
<tr>
<td>5</td>
<td>RYB</td>
<td>R7002</td>
<td>206</td>
<td>3</td>
<td>1.44</td>
</tr>
<tr>
<td>6</td>
<td>RYB</td>
<td>R7003</td>
<td>184</td>
<td>1</td>
<td>.55</td>
</tr>
<tr>
<td>7</td>
<td>RYB</td>
<td>R7004</td>
<td>155</td>
<td>6</td>
<td>5.11</td>
</tr>
<tr>
<td>8</td>
<td>UCR</td>
<td>UCR3001</td>
<td>168</td>
<td>5</td>
<td>2.99</td>
</tr>
<tr>
<td>9</td>
<td>UCR</td>
<td>UCR3002</td>
<td>162</td>
<td>3</td>
<td>1.88</td>
</tr>
<tr>
<td>10</td>
<td>UCR</td>
<td>UCR3003</td>
<td>90</td>
<td>6</td>
<td>6.66</td>
</tr>
</tbody>
</table>

10 RECORDS
COMMHND? SAVE TO FUZE

STOP MOUNT TAPE TO SAVE DATABASE-TYPE CONTINUE
:CONTINUE
COMMHND? REPLACE MFR WITH MFR+’S’
COMMHND? MERGE ON MFR FROM FUZE

STOP MOUNT TAPE AND TYPE CONTINUE
:CONTINUE
COMMHND? LIST MFR

<table>
<thead>
<tr>
<th>RECNO</th>
<th>MFR</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>HP</td>
</tr>
<tr>
<td>2</td>
<td>HP</td>
</tr>
<tr>
<td>3</td>
<td>HP</td>
</tr>
<tr>
<td>4</td>
<td>HPS</td>
</tr>
<tr>
<td>5</td>
<td>HPS</td>
</tr>
<tr>
<td>6</td>
<td>HPS</td>
</tr>
<tr>
<td>7</td>
<td>RYB</td>
</tr>
<tr>
<td>8</td>
<td>RYB</td>
</tr>
<tr>
<td>9</td>
<td>RYB</td>
</tr>
<tr>
<td>10</td>
<td>RYB</td>
</tr>
<tr>
<td>11</td>
<td>RYBS</td>
</tr>
<tr>
<td>12</td>
<td>RYBS</td>
</tr>
<tr>
<td>13</td>
<td>RYBS</td>
</tr>
<tr>
<td>14</td>
<td>RYBS</td>
</tr>
<tr>
<td>15</td>
<td>UCR</td>
</tr>
<tr>
<td>16</td>
<td>UCR</td>
</tr>
<tr>
<td>17</td>
<td>UCR</td>
</tr>
<tr>
<td>18</td>
<td>UCRS</td>
</tr>
<tr>
<td>19</td>
<td>UCRS</td>
</tr>
<tr>
<td>20</td>
<td>UCRS</td>
</tr>
</tbody>
</table>

20 RECORDS

Figure 4. Example of RECALL run for SORT and MERGE commands.
```
COMMAND? SIZE
20 RECORDS
COMMAND? DELETE FOR MFR: I
COMMAND? LIST.

<table>
<thead>
<tr>
<th>RECNO</th>
<th>MFR</th>
<th>LOT</th>
<th>SIZE</th>
<th>REJECTS</th>
<th>REJ. SIZE</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>HP</td>
<td>HP001</td>
<td>210</td>
<td>3</td>
<td>1.44</td>
</tr>
<tr>
<td>2</td>
<td>HP</td>
<td>HP002</td>
<td>206</td>
<td>2</td>
<td>0.99</td>
</tr>
<tr>
<td>3</td>
<td>HP</td>
<td>HP003</td>
<td>106</td>
<td>5</td>
<td>4.77</td>
</tr>
<tr>
<td>4</td>
<td>HPS</td>
<td>HP001</td>
<td>210</td>
<td>3</td>
<td>1.44</td>
</tr>
<tr>
<td>5</td>
<td>HPS</td>
<td>HP002</td>
<td>206</td>
<td>2</td>
<td>0.99</td>
</tr>
<tr>
<td>6</td>
<td>HPS</td>
<td>HP003</td>
<td>106</td>
<td>5</td>
<td>4.77</td>
</tr>
</tbody>
</table>

6 RECORDS
COMMAND? APPEND FROM FUZE
STOP MOUNT TAPE AND TYPE CONTINUE
: CONTINUE
COMMAND? LIST MFR

<table>
<thead>
<tr>
<th>RECNO</th>
<th>MFR</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>HP</td>
</tr>
<tr>
<td>2</td>
<td>HP</td>
</tr>
<tr>
<td>3</td>
<td>HP</td>
</tr>
<tr>
<td>4</td>
<td>HPS</td>
</tr>
<tr>
<td>5</td>
<td>HPS</td>
</tr>
<tr>
<td>6</td>
<td>HPS</td>
</tr>
<tr>
<td>7</td>
<td>HP</td>
</tr>
<tr>
<td>8</td>
<td>HP</td>
</tr>
<tr>
<td>9</td>
<td>HP</td>
</tr>
<tr>
<td>10</td>
<td>RYB</td>
</tr>
<tr>
<td>11</td>
<td>RYB</td>
</tr>
<tr>
<td>12</td>
<td>RYB</td>
</tr>
<tr>
<td>13</td>
<td>RYB</td>
</tr>
<tr>
<td>14</td>
<td>UCR</td>
</tr>
<tr>
<td>15</td>
<td>UCR</td>
</tr>
<tr>
<td>16</td>
<td>UCR</td>
</tr>
</tbody>
</table>

16 RECORDS

Figure 5. Example of RECALL run for SIZE, DELETE, and TAPE APPEND commands.
COMMAND? STRUCTURE

<table>
<thead>
<tr>
<th>FIELD</th>
<th>TYPE</th>
<th>WIDTH</th>
<th>NAME</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>C</td>
<td>5</td>
<td>MFR</td>
</tr>
<tr>
<td>2</td>
<td>C</td>
<td>10</td>
<td>LOT</td>
</tr>
<tr>
<td>3</td>
<td>I</td>
<td>4</td>
<td>SIZE</td>
</tr>
<tr>
<td>4</td>
<td>I</td>
<td>7</td>
<td>REJECTS</td>
</tr>
<tr>
<td>5</td>
<td>N</td>
<td>8</td>
<td>REJ. SIZE</td>
</tr>
</tbody>
</table>

COMMAND? 1:2:3:10:16 SAVE TO FUZE

STOP MOUNT TAPE TO SAVE DATABASE-TYPE CONTINUE

CONTINUE

COMMAND? LOAD FUZE

STOP MOUNT TAPE AND TYPE CONTINUE

CONTINUE

DATABASE FUZE LAST SAVED 8/22/75 HAS BEEN LOADED.

COMMAND? LIST

<table>
<thead>
<tr>
<th>RECNO</th>
<th>MFR</th>
<th>LOT</th>
<th>SIZE</th>
<th>REJECTS</th>
<th>REJ SIZE</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>HP</td>
<td>HP001</td>
<td>210</td>
<td>3</td>
<td>1.44</td>
</tr>
<tr>
<td>2</td>
<td>HP</td>
<td>HP002</td>
<td>206</td>
<td>2</td>
<td>.99</td>
</tr>
<tr>
<td>3</td>
<td>HP</td>
<td>HP003</td>
<td>106</td>
<td>5</td>
<td>4.77</td>
</tr>
<tr>
<td>4</td>
<td>RYB</td>
<td>R7001</td>
<td>200</td>
<td>2</td>
<td>1.00</td>
</tr>
<tr>
<td>5</td>
<td>RYB</td>
<td>R7002</td>
<td>206</td>
<td>3</td>
<td>1.44</td>
</tr>
<tr>
<td>6</td>
<td>RYB</td>
<td>R7003</td>
<td>184</td>
<td>1</td>
<td>.55</td>
</tr>
<tr>
<td>7</td>
<td>RYB</td>
<td>R7004</td>
<td>95</td>
<td>2</td>
<td>2.11</td>
</tr>
<tr>
<td>8</td>
<td>UCR</td>
<td>UCR3001</td>
<td>168</td>
<td>5</td>
<td>2.99</td>
</tr>
<tr>
<td>9</td>
<td>UCR</td>
<td>UCR3002</td>
<td>162</td>
<td>3</td>
<td>1.88</td>
</tr>
<tr>
<td>10</td>
<td>UCR</td>
<td>UCR3003</td>
<td>90</td>
<td>6</td>
<td>6.66</td>
</tr>
</tbody>
</table>

10 RECORDS

COMMAND? COUNT FOR REJECTS > 1 AND MFR = 'RYB'

3 RECORDS

COMMAND? AVERAGE REJ. SIZE; REJECTS

REJ SIZE; REJECTS

2.383 ; 3.2

10 RECORDS

Figure 6. Example of RECALL run for STRUCTURE, LOAD, COUNT, and AVERAGE commands.
COMMAND? REPORT

STOP -MOUNT REPORT TAPE AND TYPE CONTINUE
:CONTINUE
REPORT OUTPUT TO? T
REPORT FORM NAME? FUZE RECORD
UPDATE REPORT FORM? YES
HEADING? YES
DOUBLE SPACE? NO
TOTALS? YES
SUBTOTALS? YES
BY ITEMS? MFR
SUMMARY REPORT ONLY? NO
1 WIDTH; CONTENTS? 6; MFR
2 WIDTH; CONTENTS? 10; LOT
3 WIDTH; CONTENTS? 4; SIZE
4 WIDTH; CONTENTS? 7; REJECTS
5 WIDTH; CONTENTS? 8; (REJECTS/SIZE)*100
6 WIDTH; CONTENTS?
COL HEADING
1 ? MANU-/FACT
2 ? LOT NAME
3 ? SIZE
4 ? REJECTS
5 ? %/REJECTS
COLUMNS-TOTALS; NO OF DECIMAL PLACES
3 ? YES; 0
4 ? YES; 0
5 ? NO; 2

STOP SELECT PRINTER AND TYPE CONTINUE
:CONTINUE

Figure 7. Example of REPORT command.
<table>
<thead>
<tr>
<th>MFR</th>
<th>LOT NAME</th>
<th>SIZE</th>
<th>REJECTS</th>
<th>% REJECTS</th>
</tr>
</thead>
<tbody>
<tr>
<td>HP</td>
<td>HP001</td>
<td>210</td>
<td>3</td>
<td>1.44</td>
</tr>
<tr>
<td>HP</td>
<td>HP002</td>
<td>206</td>
<td>2</td>
<td>.99</td>
</tr>
<tr>
<td>HP</td>
<td>HP003</td>
<td>106</td>
<td>5</td>
<td>4.77</td>
</tr>
<tr>
<td></td>
<td><strong>TOTAL FOR MFR HP</strong></td>
<td></td>
<td>522</td>
<td>10</td>
</tr>
<tr>
<td>RYB</td>
<td>R7001</td>
<td>200</td>
<td>2</td>
<td>1.00</td>
</tr>
<tr>
<td>RYB</td>
<td>R7002</td>
<td>206</td>
<td>3</td>
<td>1.44</td>
</tr>
<tr>
<td>RYB</td>
<td>R7003</td>
<td>184</td>
<td>1</td>
<td>.55</td>
</tr>
<tr>
<td>RYB</td>
<td>R7004</td>
<td>95</td>
<td>2</td>
<td>2.11</td>
</tr>
<tr>
<td></td>
<td><strong>TOTAL FOR MFR RYB</strong></td>
<td></td>
<td>685</td>
<td>8</td>
</tr>
<tr>
<td>UCR</td>
<td>UCR3001</td>
<td>168</td>
<td>5</td>
<td>2.99</td>
</tr>
<tr>
<td>UCR</td>
<td>UCR3002</td>
<td>162</td>
<td>3</td>
<td>1.88</td>
</tr>
<tr>
<td>UCR</td>
<td>UCR3003</td>
<td>90</td>
<td>6</td>
<td>6.66</td>
</tr>
<tr>
<td></td>
<td><strong>GRAND TOTAL</strong></td>
<td></td>
<td>1627</td>
<td>32</td>
</tr>
</tbody>
</table>

Figure 8. Example of REPORT command continued.
COMMAND? REPORT

STOP -MOUNT REPORT TAPE AND TYPE CONTINUE
:CONTINUE
REPORT OUTPUT TO? FILE
REPORT FORM NAME? BLANK
UPDATE REPORT FORM? YES
HEADING? NO
DOUBLE SPACE? NO
TOTALS? NO
1 WIDTH: CONTENTS? 5; MFR
2 WIDTH: CONTENTS? 10; SIZE
3 WIDTH: CONTENTS?
COLUMNS- NO OF DECIMAL PLACES
2 ? 0

STOP LOAD DATA TAPE AND TYPE CONTINUE
:CONTINUE
COMMAND? CREATE TEST
PLEASE TYPE IN STRUCTURE OF DATA BASE

NAME; WIDTH; TYPE; DECPL
1 ? MFR; 5; C
2 ? SIZE; 10; I
3 ?

MFR    ; SIZE
?     0 RECORDS
COMMAND? APPEND FROM FILE

STOP MOUNT TAPE AND TYPE CONTINUE
:CONTINUE
COMMAND? LIST

RECNO MFR   SIZE
   1 HP    210
   2 HP    206
   3 HP    106
   4 RYB   200
   5 RYB   206
   6 RYB   184
   7 RYB   95
   8 UCR   168
   9 UCR   162
  10 UCR   90

10 RECORDS
COMMAND? QUIT

STOP -PROGRAM COMPLETED

Figure 9. Example of REPORT command to generate tape.
APPENDIX A.—USING RECALL DATA BASE IN BASIC PROGRAM

<table>
<thead>
<tr>
<th>Section</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>SUMMARY</td>
<td>34</td>
</tr>
<tr>
<td>Figure 1. Listing of ACCESS</td>
<td>35</td>
</tr>
</tbody>
</table>
It is possible to take the RECALL data base stored on tape and, with a call to a subroutine, retrieve the fields for each record in the data base. Hence, the user can write his own BASIC programs to manipulate the data base that was originally created using RECALL. The two routines are called DEFFN'81 and DEFFN'82. They appear in lines 9001 to 9011 in the file ACCESS. There is also a dimension statement in line 1. The user loads the file and then adds the appropriate statements to define the operations he wishes to perform.

The user calls DEFFN'81 once to set up the data tape and read in the structured information. The definition of each variable is given in the program listing (fig. A-1). The user then calls DEFFN'82 each time a new record is desired. The system returns the set of fields in the array W8$. When it reaches zero, all the records have been read.

For example, read in the data base that contains numeric information in fields 3 and 4 that should be checked to determine how many records have identical values in the two fields.

```
LOADDCF "ACCESS"
10 REM** I IS COUNT
20 REM SET UP TAPE: GOSUB'81
30 IF U8=0 THEN 90
40 REM READ RECORD: GOSUB'82
50 CONVERT W8$(3) TO X: CONVERT W8$(4) TO Y
60 IF X<>Y THEN 30
70 REM FIELDS ARE SAME
80 I=I+1: GOTO 30
90 PRINT "COUNT ON RECORDS WITH IDENTICAL FIELDS 3 and 4-";I
100 STOP
```
APPENDIX A

Figure A-1. Listing of ACCESS.
APPENDIX B.—LISTING OF SOURCE CODE

SUMMARY

FIGURES

B-1 Listing of RECALL
B-2 Listing of RETR1
B-3 Listing of RETR2
B-4 Listing of RETR3
B-5 Listing of RETR4

TABLES

B-I Variable Definitions
B-II Precedence Table for Conditions
B-III Subroutine Definitions
Appendix B gives a brief description of the program developed to implement the RECALL system. The database is organized as a sequence of 20 byte blocks. Each record has its fields concatenated with no separating characters. However, every record uses an integral number of blocks. Hence, if the actual number of characters in the record is 61, four blocks are used, but if the number is 60, then three blocks are used. There are 12 blocks to a sector, i.e., the space permitted for a Wang physical record (max 253 bytes). At any time, five sectors are in memory and are called a page. The remaining sectors are saved in temporary storage on the removable disk (line 2080 to 2399).

A description of the important variables in the program is given in table B-I. The precedence table used for evaluating conditions is given in table B-II.

The main program is stored in the file RECALL. It handles the general structure of a command and contains all the general-purpose subroutines. The file RETR1 handles the commands QUIT, CREATE, APPEND, LOAD, SAVE, and CHANGE. The file RETR2 handles the commands MERGE and SORT. The file RETR3 handles the commands LIST, PRINT, FAST, STRUCTURE, SIZE, SUM, AVERAGE, COUNT, REPLACE, and DELETE. The file RETR4 handles the command REPORT. The definitions of all the subroutines are given in table B-III. The listing of all the program files appears in figures B-1 to B-5.
### APPENDIX B

**TABLE B-1. VARIABLE DEFINITIONS**

<table>
<thead>
<tr>
<th>Name</th>
<th>Meaning</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Data base format</strong></td>
<td></td>
</tr>
<tr>
<td>R$1$-$60$</td>
<td>Page of blocks</td>
</tr>
<tr>
<td>R1</td>
<td>Present block index within page</td>
</tr>
<tr>
<td>R0</td>
<td>Present page index in memory</td>
</tr>
<tr>
<td>R2</td>
<td>Number of blocks allowed in memory (60)</td>
</tr>
<tr>
<td>R3</td>
<td>Last-used character index in present block</td>
</tr>
<tr>
<td>R4</td>
<td>Number of sectors per page (five)</td>
</tr>
<tr>
<td><strong>Structure file</strong></td>
<td></td>
</tr>
<tr>
<td>F5</td>
<td>File name</td>
</tr>
<tr>
<td>F4</td>
<td>Number of blocks in file</td>
</tr>
<tr>
<td>F10</td>
<td>File date</td>
</tr>
<tr>
<td>F1</td>
<td>Number of blocks per record in file</td>
</tr>
<tr>
<td>F$1$-$17$</td>
<td>Field format for record (bytes 1 to 8: name, 9: width in BIN, 10: type (I,C,N), 11: number of decimals in BIN)</td>
</tr>
<tr>
<td>F0</td>
<td>Number of fields in record</td>
</tr>
<tr>
<td><strong>Range list</strong></td>
<td></td>
</tr>
<tr>
<td>R6</td>
<td>Number of range pairs</td>
</tr>
<tr>
<td>R0(1-5)</td>
<td>Starting record index</td>
</tr>
<tr>
<td>R1(1-5)</td>
<td>Ending record index</td>
</tr>
<tr>
<td><strong>General</strong></td>
<td></td>
</tr>
<tr>
<td>C0</td>
<td>Command index</td>
</tr>
<tr>
<td>BS(1-15)</td>
<td>Field values for record</td>
</tr>
<tr>
<td>L0</td>
<td>Number of fields in field list</td>
</tr>
<tr>
<td>LS15</td>
<td>Field index table for list</td>
</tr>
<tr>
<td>C</td>
<td>Count on I/O buffer</td>
</tr>
<tr>
<td>CS(1-12)</td>
<td>I/O buffer for records</td>
</tr>
<tr>
<td>C1S(1-12)</td>
<td>I/O buffer for records</td>
</tr>
<tr>
<td><strong>FOR condition</strong></td>
<td></td>
</tr>
<tr>
<td>DS(1-12)</td>
<td>Condition string stack</td>
</tr>
<tr>
<td>PS(1-80)</td>
<td>Condition operator stack</td>
</tr>
<tr>
<td>D0</td>
<td>Number entries in condition stack</td>
</tr>
<tr>
<td>D1</td>
<td>Value of condition (1: true, 0: false)</td>
</tr>
<tr>
<td>D2</td>
<td>Position of first condition stack entry following FOR</td>
</tr>
<tr>
<td>E4</td>
<td>Number of string constants in stack</td>
</tr>
<tr>
<td>E5</td>
<td>Number of numerical constants in stack</td>
</tr>
<tr>
<td>E2</td>
<td>Number of string expressions less constant in string stack</td>
</tr>
<tr>
<td>E1</td>
<td>Number of numerical expressions less constant in number stack</td>
</tr>
<tr>
<td>E(1-10)</td>
<td>Condition number stack</td>
</tr>
<tr>
<td>T5(1-10)</td>
<td>Temporary terminal stack</td>
</tr>
<tr>
<td>E3</td>
<td>Number of terminals in temporary stack</td>
</tr>
</tbody>
</table>
### APPENDIX B

#### TABLE B-1. VARIABLE DEFINITIONS (Cont'd)

<table>
<thead>
<tr>
<th>Name</th>
<th>Meaning</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>SORT-MERGE</strong></td>
<td></td>
</tr>
<tr>
<td>D$ (1-12)</td>
<td>Output buffer to temporary disk file</td>
</tr>
<tr>
<td>B$ (1-3)</td>
<td>Field values for first file</td>
</tr>
<tr>
<td>BS (6-8)</td>
<td>Field values for second file</td>
</tr>
<tr>
<td>E4</td>
<td>Number of pages for command SORT</td>
</tr>
<tr>
<td>Q</td>
<td>Number of records in present page to sort</td>
</tr>
<tr>
<td>R0 (1-2)</td>
<td>Index of last record for files 1 and 2</td>
</tr>
<tr>
<td>E (6-7)</td>
<td>Index of record processed for files 1 and 2</td>
</tr>
<tr>
<td>E (9-10)</td>
<td>Fetch indicator for files 1 and 2 (1: fetch)</td>
</tr>
<tr>
<td>R0 (3-4)</td>
<td>Block index for files 1 and 2</td>
</tr>
<tr>
<td>E (4-5)</td>
<td>Index of sector for files 1 and 2</td>
</tr>
<tr>
<td>G (1-3)</td>
<td>Block index for field in record</td>
</tr>
<tr>
<td>E (1-3)</td>
<td>Character position for field in record</td>
</tr>
<tr>
<td>R1 (1-3)</td>
<td>Length of field</td>
</tr>
</tbody>
</table>

| **Report variables**                                    |                                                                          |
| HS        | Data base name                                                          |
| ES        | Report form name                                                        |
| GS (1)    | Heading indicator                                                       |
| GS (2)    | Double space indicator                                                  |
| GS (3)    | Totals indicator                                                        |
| GS (4)    | Subtotals indicator                                                     |
| GS (5)    | Summary report indicator                                                |
| G (1-5)   | Index of subtotal item field                                            |
| X (1-16)  | Width of report column                                                  |
| X (1-15)  | Pointer to first position in condition stack for expression             |
| SS (15,3) | Heading table for columns                                               |
| HS (1-15) | Report column information (byte 1: type C, I, N, 2: total Y, N, 3: number of decimal places in BIN) |
| T1        | Page count                                                              |
| T2        | Line count                                                              |
| T3        | Number of columns                                                       |
| T4        | Number of items                                                         |
| X (5,1-3) | Column subtotals                                                        |
| X (5,4)   | Column grand total                                                      |
| M$ (1-3)  | Value of subtotal item field                                            |
| X (1-3)   | Pointer to last position in condition stack before column contents information (i.e., last position of FOR condition) |

40
### Table B-11. Precedence Table for Conditions

<table>
<thead>
<tr>
<th>Code</th>
<th>Precedence</th>
<th>Name</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>0</td>
<td>Numeric field name</td>
</tr>
<tr>
<td>2</td>
<td>0</td>
<td>Numeric constant</td>
</tr>
<tr>
<td>3</td>
<td>0</td>
<td>Character field name</td>
</tr>
<tr>
<td>4</td>
<td>0</td>
<td>Character constant</td>
</tr>
<tr>
<td>5</td>
<td>15-0*</td>
<td>(</td>
</tr>
<tr>
<td>6</td>
<td>0</td>
<td>)</td>
</tr>
<tr>
<td>7</td>
<td>13</td>
<td>+</td>
</tr>
<tr>
<td>8</td>
<td>12</td>
<td>/</td>
</tr>
<tr>
<td>9</td>
<td>12</td>
<td>=</td>
</tr>
<tr>
<td>10</td>
<td>11</td>
<td>-</td>
</tr>
<tr>
<td>11</td>
<td>11</td>
<td>&lt;</td>
</tr>
<tr>
<td>12</td>
<td>10</td>
<td>&gt;</td>
</tr>
<tr>
<td>13</td>
<td>10</td>
<td>=</td>
</tr>
<tr>
<td>14</td>
<td>10</td>
<td>#</td>
</tr>
<tr>
<td>15</td>
<td>10</td>
<td>&lt;=</td>
</tr>
<tr>
<td>16</td>
<td>10</td>
<td>&gt;=</td>
</tr>
<tr>
<td>17</td>
<td>10</td>
<td>NOT</td>
</tr>
<tr>
<td>18</td>
<td>9</td>
<td>AND</td>
</tr>
<tr>
<td>19</td>
<td>8</td>
<td>OR</td>
</tr>
<tr>
<td>20</td>
<td>7</td>
<td>IN</td>
</tr>
<tr>
<td>21</td>
<td>10</td>
<td>NOT IN</td>
</tr>
<tr>
<td>22</td>
<td>10</td>
<td>- (Unary)</td>
</tr>
<tr>
<td>23</td>
<td>14</td>
<td>WITH</td>
</tr>
<tr>
<td>24</td>
<td>6</td>
<td>FOR</td>
</tr>
<tr>
<td></td>
<td>0</td>
<td>;</td>
</tr>
</tbody>
</table>

*Precedence is initially 15, but is stored in the stack as 0.*
## APPENDIX B

### TABLE B-III. SUBROUTINE DEFINITIONS

<table>
<thead>
<tr>
<th>Index</th>
<th>Definition</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>Reads in field from input string</td>
</tr>
<tr>
<td>2</td>
<td>Initializes data base</td>
</tr>
<tr>
<td>3(C$,N)</td>
<td>Checks for reserved word C$</td>
</tr>
<tr>
<td>4</td>
<td>Converts string to number</td>
</tr>
<tr>
<td>5(N)</td>
<td>Retrieves page containing record N</td>
</tr>
<tr>
<td>6</td>
<td>Loads data base tape</td>
</tr>
<tr>
<td>7</td>
<td>Fetches record and checks for condition (14=1 indicates record is to be used)</td>
</tr>
<tr>
<td>8(M)</td>
<td>Stores field in record</td>
</tr>
<tr>
<td>9(N)</td>
<td>Retrieves field index from field-list table</td>
</tr>
<tr>
<td>10(N)</td>
<td>Stores field index in field-list table</td>
</tr>
<tr>
<td>12(M)</td>
<td>Retrieves field in record</td>
</tr>
<tr>
<td>13</td>
<td>Retrieves record</td>
</tr>
<tr>
<td>14</td>
<td>Stores record</td>
</tr>
<tr>
<td>16</td>
<td>Stores FOR condition on stack</td>
</tr>
<tr>
<td>19(X1,X2)</td>
<td>Evaluates condition stack over indices X1-X2</td>
</tr>
<tr>
<td>20</td>
<td>Stores data-base structure for field</td>
</tr>
<tr>
<td>21</td>
<td>Reads and stores data-base field value after validation</td>
</tr>
<tr>
<td>23</td>
<td>Reads and stores field list</td>
</tr>
<tr>
<td>25</td>
<td>Adds constant or field to evaluation stacks</td>
</tr>
<tr>
<td>26</td>
<td>Checks for string in string</td>
</tr>
<tr>
<td>28</td>
<td>Retrieves expression as string for updating field</td>
</tr>
<tr>
<td>29(Y)</td>
<td>Converts number to string in proper format</td>
</tr>
<tr>
<td>31(X)</td>
<td>Saves page onto disk</td>
</tr>
</tbody>
</table>
TABLE B-III. SUBROUTINE DEFINITIONS (Cont'd)

<table>
<thead>
<tr>
<th>Index</th>
<th>Definition</th>
</tr>
</thead>
<tbody>
<tr>
<td>34</td>
<td>Transfers temporary disk file onto active disk file</td>
</tr>
<tr>
<td>37</td>
<td>Saves block onto temporary disk file</td>
</tr>
<tr>
<td>40</td>
<td>Selects temporary file for SORT-MERGE</td>
</tr>
<tr>
<td>41</td>
<td>Defines location of SORT-MERGE fields within record</td>
</tr>
<tr>
<td>42</td>
<td>Fetches field value from active file</td>
</tr>
<tr>
<td>43</td>
<td>Fetches field value from tape file</td>
</tr>
<tr>
<td>44</td>
<td>Compares active and tape file and performs merge</td>
</tr>
<tr>
<td>45(U)</td>
<td>Loads sector for file U</td>
</tr>
<tr>
<td>46(U)</td>
<td>Fetches field value from file U</td>
</tr>
<tr>
<td>47</td>
<td>Compares field from two files and performs merge</td>
</tr>
<tr>
<td>53(X,Y)</td>
<td>Stores field or constant index in conditional stack</td>
</tr>
<tr>
<td>54</td>
<td>Checks for field name (X=1 if there is a field name)</td>
</tr>
<tr>
<td>63</td>
<td>Updates character count in field retrieval</td>
</tr>
<tr>
<td>68</td>
<td>Fetches record to save in I/O buffer after checking FOR condition</td>
</tr>
<tr>
<td>69</td>
<td>Fetches record to save in I/O buffer</td>
</tr>
<tr>
<td>70</td>
<td>Fetches page from disk</td>
</tr>
<tr>
<td>71</td>
<td>Stores page onto disk</td>
</tr>
<tr>
<td>91</td>
<td>Prints page heading for report</td>
</tr>
<tr>
<td>92(J,K)</td>
<td>Prints subtotal item field value</td>
</tr>
<tr>
<td>93(K,Y)</td>
<td>Converts numerical column value to string</td>
</tr>
<tr>
<td>94(T)</td>
<td>Right justifies numerical string</td>
</tr>
<tr>
<td>95</td>
<td>Increments line count</td>
</tr>
<tr>
<td>96</td>
<td>Prints subtotal values for numerical columns</td>
</tr>
<tr>
<td>97</td>
<td>Prints column values and updates subtotals</td>
</tr>
</tbody>
</table>
Appendix B

```
1000 REM *** START OF COMMANDS
1100 REM **+ END OF COMMANDS
```

**Figure B-1. Listing of RECALL.**
APPENDIX B

1160 DEFEN 1: REM READ IN NEXT FIELD: IF C1$="""CHA"" THEN 1170: I1=1: B$=B$(I1): RETURN
1170 IF A$="""& THEN 1175: B$=A$: I1=POS(A$=""): IF I1=0 THEN 1180: IF I1=I1 THEN 1190: B$=STR(B$, I1-I1-1): GOTO 1200
1175 INPUT A$: GOTO 1170
1180 A$="": IF B$="" THEN 1210: I1=1: GOTO 1210
1190 B$="":
1200 I1=STR(A$, I1+1): I1=1
1210 RETURN
1220 DEFEN 2: F0: R1, R3, R0, F4=0: DBACKSPACE BEG : L=POS(A$=""" ""): IF L =0 THEN 1230: F$=STR(A$, L): RETURN
1230 INPUT "DATABASE", F$: RETURN
1240 DEFEN 3(C$: N)
1250 IF LEN(A$)< N THEN 1260: IF STR(A$, I1, N)="": THEN 1270: A$=STR(A$, 2): GOTO 1250
1260 N=0: RETURN
1270 IF A$=STR(A$, N+1): A$=STR(A$, POS(A$=""" ""): RETURN
1280 DEFEN 4: REM CONVERT TO NUMBER
1290 IF B$="" THEN 1300: IF NUM(B$)<0 THEN 1310: IF NUM(B$)<LE
1300 PRINT "ILLEGAL NUMBER" 
1310 RETURN
1320 DEFEN 5: STOP "MOUNT TAPE AND TYPE CONTINUE": IF C0=3 THEN 1325 : DATA LOAD "STRUCT": DATA LOAD G$, I9: H$, F1, F$: F0: GOTO 1327
1325 DATA LOAD "STRUCT": DATA LOAD G$, I9
1327 IF A$=G$ THEN 1330: PRINT "WRONG DATABASE ": I$=I9=0: RETURN
1330 DATA LOAD "FILE": RETURN
1340 DEFEN 8(N): REM STORE FIELD IN RECORD: K=1
1350 IF R3>21 THEN 1370: R1=R1+1: R3=1: IF R1<=R2 THEN 1370: GOSUB 63: STR(C$, R1, 18) =STR(C$: M), B$: I7 =I7+1: RETURN
1360 RETURN
1370 GOSUB 63: STR(R$, R1, 18) =STR(B$: K, I9): K=K+I9: R3=R3+I8-1: I7=I7+1: RETURN
1380 DEFEN 12(M): REM RETRIEVE FIELD IN RECORD: B$=""" ": K=1
1390 IF R3>21 THEN 1440: R3=1: R1=R1+1: IF R1<=R2 THEN 1420: GOSUB 70: R0=R0+1: R1=1
1400 SUB: K=K+1: IF C<12 THEN 1430: C$(C)=R$(R1): GOTO 1440
1410 IF R3>21 THEN 1440: R3=1: R1=R1+1: IF R1<=R2 THEN 1420: GOSUB 70: R0=R0+1: R1=1
1420 C=0+1: IF C12 THEN 1430: C$(C)=R$(R1): GOTO 1440
1430 IF (C<12) THEN 1410: RETURN
1440 GOSUB 63: STR(R$, K, I8) =STR(R$(R1), R3, I8): R3=R3+I8-1: K=K+I8: RETURN
1450 DEFEN 13: REM RETRIEVE RECORD
1460 FOR I7=1 TO F0: GOSUB 12(VAL(STR(F$(I7)), 9)): B$(I7)=B$: NEXT I7
1470 RETURN
1480 DEFEN 14: REM STORE RECORD
1490 FOR I7=1 TO F0: B$=B$(I7): GOSUB 8(VAL(STR(F$(I7)), 9)): NEXT I7
1500 RETURN
1510 DEFEN 5(N): REM INITIATE FETCH: N=(N-1)*F1+1
1520 IF N<1 THEN 1525: DSKIP F5=R4$: GOSUB 70: R0=F5: GOTO 1530
1530 R0=0: R1=-1
1540 RETURN
1540 DEFEN 9(N): L=VAL(STR(L$, N)): RETURN
1550 DEFEN 10(N): BIN(STR(L$, N))=1: RETURN

Figure B-1. Listing of RECALL (Cont'd).
APPENDIX B

1560 DEFN'53'::X=0: IF X<>HEX<2E>THEN 1910: IF X<>HEX<30>THEN 1920: IF X<>HEX<5A>THEN 1920: IF X<>HEX<3F>THEN 1910: IF X<>HEX<3A>THEN 1910: X=1
1920RETURN

Figure B-1. Listing of RECALL (Cont'd).
IF I7>2 THEN E1=E1+1: GOTO 1950

1950: E2=E2+1

1950: I7=15+1: X=VAL(P$(I5)) : RETURN

1960: FOR I7=1 TO LEN(B$)-LEN(A$)+1: IF A$>STR$(B$, I7, LEN(A$)) THEN 1970: I7=1: RETURN

1970: NEXT I7: I7=0: RETURN

1980: FOR I7=1 TO X2: D3=0: E2=E4: E1=E5

1990: IF I7>X1 THEN 2040: GOSUB '25: ON

I7 GOTO 2000, 2010, 2020, 2030

2000: INVERT B$(X) TO E(E1) GOTO 2400

2010: E(E1)=E(X) GOTO 2400

2020: E(E2)=E(X) GOTO 2400

2030: E(E2)=E(X) GOTO 2400

2040: IF E1=E3 THEN 2060: IF E1=2 THEN 2050: X=E(E1-1)

2050: Y=E(E1)

2060: IF E2=E4 THEN 2080: IF E2=E4 THEN 2070: X=E(E2-1)

2070: E2=E2-1: GOTO 2400

2080: UN I7=6 GOTO 2090, 2100, 2110, 2120, 2130, 2140, 2150, 2160, 2170, 2180, 2190, 2200, 2210, 2220, 2230, 2240, 2250, 2260, 2270, 2280, 2290, 2300, 2310, 2320

2090: X=X: Y=GOTO 2390

2100: X+X: Y=GOTO 2390

2110: X=X: Y=GOTO 2390

2120: IF E2=E4 THEN 2130: IF X!=0 THEN 2300

2130: STR$(D$(X-1), LEN(A$)+1)=B$: E2=E2-1: GOTO 2400

2140: X=X: Y=GOTO 2390

2150: IF E2=E4 THEN 2160: IF X<Y THEN 2350: GOTO 2370

2160: A$=B$THEN 2330: GOTO 2340

2170: IF E2=E4 THEN 2180: IF X<Y THEN 2350: GOTO 2370

2180: A$=B$THEN 2330: GOTO 2340

2190: IF E2=E4 THEN 2200: IF X<Y THEN 2350: GOTO 2370

2200: IF A$=B$THEN 2330: GOTO 2340

2210: IF E2=E4 THEN 2220: IF X<Y THEN 2350: GOTO 2370

2220: A$=B$THEN 2330: GOTO 2340

2230: IF E2=E4 THEN 2240: IF X=0 THEN 2350: GOTO 2370

2240: A$=B$THEN 2330: GOTO 2340

2250: IF E2=E4 THEN 2260: IF X=0 THEN 2350: GOTO 2370

2260: IF A$=B$THEN 2330: GOTO 2340

2270: E(E1)=1-E(E1): GOTO 2400

2280: X=X+Y: GOTO 2370

2290: GOSUB '26: IF I7=1 THEN 2320: GOTO 2370

2300: GOSUB '26: IF I7=1 THEN 2320: GOTO 2370

2310: E(E1)=E(E1): GOTO 2400

2320: GOSUB '28: D3=15: GOTO 2400

2330: E(E1+1)=1: GOTO 2350

2340: E(E1+1)=0

2350: E1=E1+1: E2=E2-2: GOTO 2400

2360: E(E1-1)=1: GOTO 2360

2370: E(E1-1)=0

2380: E1=E1-1: GOTO 2400

2390: E1=E1-1: E(E1)=X

2400: NEXT I5

2410: IF E1>=ESTHEN 2420: RETURN

2420: E1=E1: RETURN

Figure B-1. Listing of RECALL (Cont'd.).
APPENDIX B

Figure B-1. Listing of RECALL (Cont'd).
APPENDIX B

Figure B-2. Listing of RETR1.
**APPENDIX B**

630 REM **APPEND.
640 GOSUB '5(F/4/F1): DBACKSPACE R2/12S: R1=R1+F1: IF A$=" " THEN 350:
GOSUB '3("FROM",4): IF N=0 THEN 220: GOSUB 6: IF I9=0 THEN 220
650 DATA LOAD C$(J): IF END THEN 710
660 FOR J=I0 TO 12
661 R1=R1+1: IF R1>R2 THEN 680: R$(R1)=C$(J): GOTO 700
680 NEXT J: GOTO 650
700 J=0: GOSUB '71: REWIND : GOTO 100
720 REM **LOAD.
730 GOSUB '73: A#=F$: GOSUB '6: IF I9=0 THEN 220
740 PRINT "DATABASE "; G$: " LAST SAVED "; H$: " HAS BEEN LOADED. ": F4 =0: GOTO 650
750 REM **SAVE.
760 GOSUB '73("TO",2): IF N=0 THEN 220
770 G$=STR(A$,1),POS(A$=" "): A#=STR(A$,POS(A$=" "): GOSUB '16: IF I6=2 THEN 220
780 STOP " MOUNT TAPE TO SAVE DATABASE-TYPE CONTINUE": REWIND : DATA SAVE OPEN "STRUCT": DATA SAVE G$, F4, F1$, F1$, F0$: DATA SAVE END : DATA SAVE OPEN "FILE": C=0
790 I9=0: FOR I=1 TO R6: GOSUB '5(R0(I)): FOR J=R0(I) TO R1(I)< R1(I)> : C=0: GOSUB '7: IF I4=0 THEN 860: FOR I4=1 TO R6: GOSUB '68(I): IF I4=1 THEN 810: C=C+F1: GOTO 820
810 I9=I9+F1: IF C<=12 THEN 820: DATA SAVE C$(J): FOR K=I TO 12: C$(K)=C$(K): NEXT K: C=C-12
820 NEXT J: NEXT I: IF N=0 THEN 830: DATA SAVE C$(J)
830 DATA SAVE END : REWIND : DATA LOAD "STRUCT": DATA RESAVE G$, I9, F4, F1$, F0$: GOTO 100
840 REM **CHANGE.
850 GOSUB '23: IF I6=2 THEN 220: FOR I2=1 TO R6: GOSUB '5(R0(I2)) FOR J=R0(I2) TO R1(I2)< R1(I2)> : C=0: GOSUB '7: IF I4=0 THEN 860: FOR I4=1 TO R6: GOSUB '68(I4): PRINT STR(F$(J),1,8): ": B$(I4): B$(I4): PRINT B$(I4): NEXT I4: R3=20: R1=R1-F1: FOR I=1 TO F0: GOSUB '2
1: NEXT I
860 GOSUB '31(I2): NEXT J: NEXT I2: GOTO 100
1081 DEFN'E69 FOR I8=1 TO F1
1082 R1=R1+1: IF R1<=R2 THEN 1083: GOSUB '70: R0=R0+1: R1=1
1083 C=C+1: IF C<=12 THEN 1084: C$(C)=R$(R1): GOTO 1085
1084 C$(C)=R$(R1)
1085 NEXT I8: RETURN
1086 DEFN'E68: I4=1: IF D2=0 THEN 1087: GOSUB '7: RETURN
1087 GOSUB '69
1088 RETURN

Figure B-2. Listing of RETRl (Cont'd.)
DATA LOAD DC OPEN RTMP ,2080,2399
2450N C0=16GOTO 250,500
250REM **MERGE:GOSUB '3"ON",2):IF N=THEN 220:GOSUB '23:IF I6=2
THEN 220:IF D2>D0+1THEN 220:GOSUB '6:IF I9=THEN 220:GOSUB '40:
260GOSUB '41
260DATA LOAD C1$:DATA LOAD C1$:GOSUB '5(I):I1=F4/F1:I2=I9/F1:
X1,M2=1:R1.I7=1-F1:18=0
270IF (I1+1)*X1=THEN 280:GOSUB '42
280IF (I2+1)*X2=THEN 290:GOSUB '43
300IF I1+I2+2=THEN 300:GOSUB '44:GOTO 270
300I9=(F4+19)/F1:DATA SAVE DC #2,2*():GOSUB '34:REWORD:GOTO 100
310DEFFN'40:SELECT #2310:DATA SAVE DC OPEN F#2,TEMP ,2080,2399:R
320RETURN
320DEFFN'41:IF L0<THEN 330:L0=I
330FOR I=1TO L0+GOSUB '9(I):X=1:IF L=1THEN 340:FOR J=1TO L-1:X=X+
VAL(STR(F$(J),9)):NEXT J
340E(I)=X-1-INT((X-1)*.05)*20:G(I)=INT((X-1)*.05):R1(I)=VAL(STR(
K$(L),9)):NEXT I:RETURN
350DEFFN'42:I1=I1-1:IF I1<THEN 375:R1=R1+F1:IF R1<=R2THEN 360:G
360S=70:R1=I
360FOR I=1TO L0:K=R1+6(G(I)):R3=E(I):B$(I)="":M=R1(I):J=I
370R3=R3+1:IF R3<2THEN 370:R3=1:K=K+1
370L=M:IF R3+M-1<2THEN 373:L=21-R3
373M=M-L:STR(B$(I),J,L)=STR(R$(K),R3,0):R3=R3+L-1:J=J+L:IF M<0TH
380EN 365:NEXT I:RETURN
390M='I(1)="":RETURN
380DEFFN'43:I2=I2-1:IF I2<THEN 425:I7=I7+F1:IF I7<13THEN 390:FO
R I=1TO 12:C$(I)=C1$:NEXT 'I:DATA LOAD C1$:I7=I7-12
420FOR I=1TO L0:K=I7+G6(I)):R3=E(I):B$(I+5)="":M=R1(I):J=I
430IF R3+1:IF R3<2THEN 400:R3=1:K=K+1
440L=M:IF R3+M-1<2THEN 405:L=21-R3
405M=M-L:IF K<13THEN 410:STR(B$(I+5),J,L)=STR(C1$(K-12),R3,0):GO
TO 420
410STR(B$(I+5),J,L)=STR(C$(K),R3,0)
420R3=R3+L-1:J=J+L:IF M<0THEN 395:NEXT I:RETURN
420M='I(6)="":RETURN
430DEFFN'44:FOR I=1TO L0:IF B$(I)>B$(I+5)THEN 460:IF B$(I)<B$(I+5)
THEN 440:NEXT I
440X1=0:X2=0:FOR J=R1TO R1+F1-1;I8=I8+1:IF I8<13THEN 450:DATA SA
440V. DC #2.D$(I):I8=1
450D$(I)=R$(J):NEXT J:RETURN
450X1=0:X2=1:FOR J=R1TO I7+F1-1;I8=I8+1:IF I8<13THEN 470:DATA SA
440V. DC #2.D$(I):I8=1
470IF J<13THEN 480:D$(I)=C1$(J-12):GOTO 490
480D$(I)=C$(J)
490NEXT J:RETURN

Figure B-3. Listing of RETR2.
Figure B-3. Listing of RETR2 (Cont'd)
APPENDIX B

Figure B-4. Listing of RETR3.
APPENDIX B

Figure B-5. Listing of RETR4.
APPENDIX B

Figure B-5. Listing of RETR4 (Cont'd).

530REM *INIT:T1=0:M$(1)=""":FOR J=1TO 4:FOR I=1TO 5:X4(I,J)=0:  
NEXT I:NEXT J:IF T4=0THEN 540:X5$(1)="*";X5$(2)="**":X5$(3)="***":  
540IF G$(1)<"Y"THEN 560:GOSUB '91  
560FOR S1=1TO R6:GOSUB '5(R0(S1)):IF R2=R0(S1):GOSUB '91(S1):GOSUB '92(S1):GOSUB '93(S1):GOSUB '94(S1):S1=0:IF N4=0THEN 610:GOSUB '93(S1):  
570IF T4=0THEN 600:IF M$(1)<"Y":GOSUB '91  
T2=T2+1:GOTO 600  
580L1=T4+1:FOR L=L0+T4+1STEP -1:K=G(L0):IF M$(L0)=B$(K):THEN 590:  
L0=L1-1:PRINT :PRINT X5$(L0);"TOTAL FOR ";STR(T2<CL,0>):"**":;B$(L0):GOSUB '96  
590NEXT L0:PRINT :GOSUB '95:IF L1>T4:GOTO 600  
600S=20:GOSUB '97  
610NEXT S2:NEXT S1:IF G*'.'T><"Y"THEN 615:PRINT "** GRAND TOTAL **";L=4:GOSUB '96  
612SELECT PRINT 00564):GOTO 100  
615IF H$="T":GOTO 612:DATA SAVE C1$():DATA SAVE END :REWRIND:DATA  
LOAD "STRUCT":FOR I=1TO T3:8$(I)=F$(I):S1$1=F$(I):STR(F$(I),1,8):"":;STR(F$(I),11)=STR(H$(I),3)  
617IF STR(F$(I),1,8)=S$(I,1):NEXT I:DATA RESAVE H$,19,F1$,F7,F$(I),T3:  
REWRIND:FOR I=1TO T3:8$(I)=B$(I):NEXT I:GOTO 100  
620DEFFN '91:G1=T1+1:PRINT HEX(T2<00>):PRINT T"PAGE":;T1:PRINT "DATE: ":;F1$:PRINT "DATABASE: ":;F$:PRINT "REPORT FORM: ":;E$:PRINT :;T2=11  
630IF G$(1)<"N":GOTO 640:FOR J=1TO 3:L=0:FOR I=1TO T3:L=L+VAL(S2$<I>)++;PRINT S2$(I,J);TAB(L));NEXT I:PRINT :NEXT J:PRINT  
640RETURN  
650DEFFN '92(J,K):PRINT X5$(J);"":;STR(F$(K),1,8);":"::;B$(K):T2=T  
2+1:RETURN  
660DEFFN '93(K,Y);N=VAL(STR(H$(K),3)) :GOSUB '29(Y):GOSUB '94(K):F  
RETURN  
710DEFFN '94(T,C):"":;STR(C$,1)+VAL(STR(X2$(T),1)-LEN(B$))=B$:B=C$:RETURN  
720DEFFN '95:IF G$(1)<"Y":GOTO 730:T2=T2+1:IF T2<56:GOTO 730:GOSUB '91  
730RETURN  
740DEFFN '96:0,P=0:FOR Z=1TO T3:IF STR(H$(Z),2,2.1)<"Y":THEN 750:P  
=P+1:GOSUB '93(Z<2,X4(P,L)):FOR I=1TO T4:4(X,P,I)=0:NEXT I:PRINT TAB  
(0):B$:  
750=0+VAL(X2$(Z)+1):NEXT Z:PRINT :GOSUB '95:RETURN  
760DEFFN '97:0,P=0:FOR I=1TO T3:GOSUB '19(VAL(X3$(I)),VAL(X3$(I)+1  
)-1):IF STR(H$(I),1,1)<"C":THEN 770:B$=D$(E2):E2=E4:GOTO 780  
770GOSUB '93(I,E(E1)):IF STR(H$(I),2,2.1)<"Y":THEN 780:P=P+1:CONVE
Figure B-5. Listing of RETR4 (Cont'd).

```
R1: B$=0 X8:X4(P,4)=X4(P,4)+X8:IF T4=0 THEN 780:FOR L=1 TO T4: X4(P, L)=X4(P, L)+X8:NEXT L
780: I F H$="T" THEN 788: M=VAL(X2$I(I)): X9=1
781: S=S+1: I F S<21 THEN 783: C1=C1+1: S=1: I F C1<13 THEN 783: DATA SAVE
C1$(): C1=1
783: I F M-I 8
784: I F S+M-I 8: 784: I 8=21-S
784: M=M-I 8
785: STR(C1$(): S, I 8)=STR(B$: X9, I 8): X9=X9+I 8: S=S+I 8-1: I F M=0 THEN
781
788: I F G$="Y":I 8=2 THEN 790: PRINT TAB(W): B$: W=W-VAL(X2$I(I))+1
790: NEXT I: I F G$="Y":I 8=2 THEN 805: PRINT : I F G$="Y":I 8=2 THEN 800: PRIN
800: GO$SUB '95: RETURN
805: RETURN
```
DISTRIBUTION (Cont'd)

J. J. GARCIA & ASSOCIATES, INC.
11039 N. E., 6TH AVENUE
MIAMI, FL  33161
ATTN ENRIQUE ALVAREZ

LINE COUPLING EQUIPMENT ENGINEERING
GENERAL ELECTRIC COMPANY
MOUNTAIN VIEW ROAD
LYNCHBURG, VA  24502
ATTN MR. D. B. BRAH, MANAGER

OREGON STATE UNIVERSITY
SCHOOL OF OCEANOGRAPHY
CORVALLIS, OR  97331
ATTN DR. LOUIS I. GORDON
ASSISTANT PROFESSOR

CARTER PRODUCTS
RESEARCH LABORATORY
CRANBURY, NJ  08512
ATTN W. M. WOODING
DIRECTOR, TECHNICAL SERVICES

PICKARD & ANDERSON ENGINEERS
69 SOUTH ST
AUBURN, NJ  13021
ATTN WILLIAM C. ANDERSON, P.E.

NORTHEASTERN PRODUCTS COMPANY
3500 S. CLINTON AVENUE
SOUTH PLAINFIELD, NJ  07080
ATTN RICHARD D. GUIDO
MANAGER QUALITY CONTROL

COMMANDANT
USA FIELD ARTILLERY SCHOOL
FORT SILL, OK  73503
ATTN BILL MILLSPAUGH
ATSPC/SD

CALCULATOR CONSULTANT
45-3A MT. PLEASANT VILLAGE
MORRIS PLAINS, NJ  07950
ATTN MR. NEAL H. KUHN

US ENVIRONMENTAL PROTECTION AGENCY
P.O. BOX 5036
ROCHESTER, NY  14627
ATTN MR. DONALD J. CASEY
CHIEF, IFYGL BRANCH

ST. MARY'S UNIVERSITY
2700 CINCINNAT AVENUE
SAN ANTONIO, TX  78284
ATTN DR. TOM NOTE

ETHYL CORPORATION
FUNDAMENTAL STUDIES DEPARTMENT
TERRE HAUTE, IN  47808
ATTN MR. CHARLES FURLAND

BOEING AEROSPACE COMPANY
P.O. BOX 3999
SEATTLE, WA  98124
ATTN MR. MALCOLM MATHENS
MS SC-41

PENNWALT CORPORATION
TECHNOLOGICAL CENTER
900 FIRST AVENUE
KING OF PRUSSIA, PA  19406
ATTN DR. J. E. DOHANY

MR. L. H. CHAMBERLIN
10510 SUNNYBROOK LANE, SW
TACOMA, WA  98498

WANG LABORATORIES
4824 BOILING BROOK PARKWAY
ROCKVILLE, MD  20852
ATTN RAY DORRIS

HARRY DIAMOND LABORATORIES
ATTN MCGRORGR, THOMAS, CO, COMMANDING
OFFICER/PYER, I.K./LANDIS, P.E./
SMITR, H./CONRAD, E.R.
ATTN CARTER, W.W., DR., ACTING TECHNICAL
DIRECTOR/ARCUS, S.M.
ATTN KIMMEL, S., IO
ATTN CHIEF, 0021
ATTN CHIEF, 0022
ATTN CHIEF, LAB 100
ATTN CHIEF, LAB 200
ATTN CHIEF, LAB 300
ATTN CHIEF, LAB 400
ATTN CHIEF, LAB 500
ATTN CHIEF, LAB 600
ATTN CHIEF, DIV 700
ATTN CHIEF, DIV 800
ATTN CHIEF, LAB 900
ATTN CHIEF, LAB 1000
ATTN RECORD COPY, BR 041
ATTN HDL LIBRARY (3 COPIES)
ATTN CHAIRMAN, EDITORIAL COMMITTEE
ATTN CHIEF, 047
ATTN TECH REPORTS, 013
ATTN PATENT LAW BRANCH, 071
ATTN MCLAUGHLIN, P.W., 741
ATTN LANHAM, C., PROGRAM & PLANS
ATTN CLASEN, S. M., 120
ATTN CHIEF, 310
ATTN MANSON, F. M., 310
ATTN DREWIECKI, T., 310
ATTN SPIRouPOoulos, C., 310
ATTN CHIEF, 340
ATTN GOTO, J., 340
ATTN MONT, G., 340
ATTN INGERSOLL, P., 430 (3 COPIES)
ATTN OVERMAN, D. L., 420
ATTN BUTLER, R., 0025
ATTN HINE, R., 0025
DISTRIBUTION (Cont'd)

HARRY DIAMOND LABORATORIES (Cont'd)
ATTN JOHNSON, P., 610
ATTN RAVILIOUS, C., 310
ATTN ROSEN, R., 800
ATTN WICKLUND, J., 280
ATTN MARROLETTI, J., 0025
ATTN MATHEWS, H. J., 0025
ATTN OFFICE, 0025 FILE
ATTN BETTWY, D., 710
ATTN FUNKE, M., 340
ATTN LIEBERMAN, S., 350
ATTN CROSBY, T., 330
ATTN DAVIDSON, H., 410
ATTN HOPP, T. H., 930
ATTN ISEMAN, J., 930
ATTN HUPMAN, A., 0025
ATTN HAUSNER, A., 0025
ATTN BLOOM, H. (30 COPIES)