An Application of Famulus: "A Bibliography of Quantitative Ecology"

by
**M. I. Cochran
J. M. Thomas**

September 1976

**Prepared for the Energy Research
and Development Administration
under Contract E(45-1)-1830**

Battelle

Pacific Northwest Laboratories

NOTICE

This report was prepared as an account of work sponsored by the United States Government. Neither the United States nor the Energy Research and Development Administration, nor any of their employees, nor any of their contractors, subcontractors, or their employees, makes any warranty, express or implied, or assumes any legal liability or responsibility for the accuracy, completeness or usefulness of any information, apparatus, product or process disclosed, or represents that its use would not infringe privately owned rights.

PACIFIC NORTHWEST LABORATORY
operated by
BATTELLE
for the
ENERGY RESEARCH AND DEVELOPMENT ADMINISTRATION
Under Contract EY-76-C-06-1830

Printed in the United States of America
Available from
National Technical Information Service
U.S. Department of Commerce
5285 Port Royal Road
Springfield, Virginia 22151
Price: Printed Copy \$____*; Microfiche \$3.00

*Pages	NTIS Selling Price
001-025	\$4.50
026-050	\$5.00
051-075	\$5.50
076-100	\$6.00
101-125	\$6.50
126-150	\$7.00
151-175	\$7.75
176-200	\$8.50
201-225	\$8.75
226-250	\$9.00
251-275	\$10.00
276-300	\$10.25

**An Application of Famulus:
"A Bibliography of
Quantitative Ecology"**

**by
M. I. Cochran
J. M. Thomas**

September 1976

Battelle
Pacific Northwest Laboratories
Richland, Washington 99352

SUMMARY

In this document we discuss how the FAMULUS System (obtained from the Pacific Southwest Forest and Range Experiment Station, P.O. Box 245, Berkeley, California, 94701) was used to build a 4000+ reference file, which was published as "A Bibliography of Quantitative Ecology" (Schultz, 1976). Examples from the main programs show the system of checks, cross-checks, information retrieval, and printouts used for editing the bibliography prior to publication. Further, we hope that the step by step presentation of the various main programs of FAMULUS, using actual examples, will both stimulate others to attempt bibliographies of their own specialized fields and show that FAMULUS is a worthwhile system to use in such an endeavor. The examples shown in this document as well as the System User's Manual should make the FAMULUS System readily understandable to interested ecologists.

CONTENTS

SUMMARY	iii
LIST OF FIGURES AND TABLES	ix
INTRODUCTION	1
GENESIS OF A DATA FILE	5
MAIN PROGRAMS	11
EDIT	11
GALLEY	29
INDEX	35
SORT	41
MERGE	53
OSSIFY	61
SEARCH	65
VOCAB	73
ACKNOWLEDGEMENTS	83
REFERENCES	85
APPENDIX A. Control Cards for Each Main Program	A. 1
APPENDIX B. Programmer's Aids and Special Programs	B. 1
APPENDIX C. Program Listings	C.1
EDIT	C. 1
GALLEY	C.8
INDEX	C. 11
MERGE	C.16
OSSIFY	C. 21
SEARCH	C.24
SORT	C. 30
VOCAB	C.37
Common Subroutines	C. 43
Adam	C.43
Arty	C. 46
Asceni	C.47
Ascens	C.48

CONTENTS (Continued)

Move	C.89
Nimrod	C.90
Noah	C.91
Pager	C.93
Polish	C.94
Pretty	C.98
Prncit	C.100
Prndic	C.102
Pundic	C.104
Rewind	C.104
Scan	C.105
Scat	C.106
Scout	C.107
Setcom	C.108
Setsym	C.111
Shell i	C.112
Shells	C.113
Shellv	C.115
Simile	C.116
Spot	C.117
Symbol	C.118
Talmud	C.121
Taper	C.122
Tinker	C.124
Usher	C.126
Vstat	C.127
APPENDIX D. Program Modifications Used for Publication of "Bibliography of Quantitative Ecology"	D.1

LIST OF FIGURES

1	Preprinted Forms for Data	7
2	FAMULUS Coding Form	9
3	Examples of Reference Cards Key punched	10
4	System and Program Control Cards for Program EDIT	12
5	Control, Command and Data Cards for Program EDIT	15
6	Output from Program EDIT	22
7	Cards Used to Add Additional Citations to an Existing Data File	25
8	Flowchart of Programs EDIT and SORT as Shown in Figure 7	26
9	Output from Programs in Figure 7	27
10	System and Program Cards Used to Run GALLEY	31
11	Output from Program GALLEY	32
12	Print Out from Program GALLEY Using Option - No Numbers	33
13	System and Program Cards Used to Run INDEX	36
14	Output from Program INDEX	36
15	Control Cards for Use with Programs EDIT and INDEX Conjointly	38
16	Output from Control Cards Shown in Figure 15	38
17	Control Cards and Flowchart for Program SORT	43
18	Control Cards for Program SORT and GALLEY	44
19	Output from Control Cards in Figure 18	45
20	Control Cards to Use Program SORT and GALLEY to Check for Errors in the DESA Field	48

LIST OF FIGURES (Continued)

D4	Two Column Format Created by Program TWOOCOLM .	. D.18
D5	Control Cards for Program INDEX with Subroutines CIPHER and XCHANGE and Data Cards for XCHANGE	. D.19
D6	Program THOMAS To Write Author Index .	. D.24
D7	Author Index Published in Bibliography .	. D.25
D8	Flow Diagram of Steps Used to Produce Text Pages and an Author Index for a Bibliography of Quantitative Ecology	. D.26

LIST OF TABLES

1	Keypunch Instructions	8
2	Logic of SEARCH Requests	66

INTRODUCTION

FAMULUS* is a computer-based system designed to aid the researcher in individual documentation activities. However, it is also structured so that an entire department could enter material on the same storage media. The system is therefore very individualized and at the same time can meet group needs. Users can structure "data" according to their specific needs. Our definition of "data" in this document refers to bibliographic citations.

The principal system features of FAMULUS are: editing, sorting, indexing, vocabulary-building, searching and file-revision. These will be discussed in detail later in this document.

Burton and Yerke (1969) give the following account of the origin of FAMULUS:

"In 1966 a computer services unit was established in the Station library. Its original function was to aid users of Station-produced and other programs and to maintain and operate the library's programs. The unit very quickly also became a support service for the growing number of personal collections. The increase in such collections began to reveal the limitations of CATALOG for this work, and the desiderata for a new system became quickly apparent. Its guiding principle had to be the greatest possible editing, sorting, deleting, revising, indexing, and searching ease for the user, within a framework of feasible input and item-size limits. Support for this work as potentially a Forest Service-wide application was received from the Washington office. The Station's senior programmer, Robert M. Russell, was given the programming assignment, working closely with the authors of this paper in the system's design. In view of the system's inclusive support features for personal documentation, it was named FAMULUS. FAMULUS is not an acronym. A Famulus was the private secretary of a medieval scholar or alchemist. Perhaps the most well-known was Wagner, Famulus of Dr. Faustus.

The general outline of FAMULUS was completed early in 1967, and programming of a FORTRAN IV version for the CDC 6400 began in June. It was completed in June 1968. All existing computer-based collections were shifted from CATALOG to FAMULUS, using an option built into the latter system. No reformatting or restructuring burdens were placed on the user."

---

*Supplied by the Pacific Southwest Forest and Range Experiment Station, Berkeley, California.

In 1974, it was decided to further subdivide the citations into more manageable categories. The CHK field was then used to insert code words as descriptors; we used one descriptor word per citation. These descriptors were abbreviations of words to be used in search formulas. The task involved considerable time because computer outputs had to be read and decisions on categorization (or 26 we devised) for each citation were made. A series of search formulas (c.f. section on search methods) were run on the entire Bibliography. Search requests, run on both the title and descriptor fields, gave us an additional index not possible with the conventional method of searching just the title fields.

At that time, we anticipated the Bibliography would be completed in March 1975. However, upon completion of a preliminary draft, the decision was made that in order for the Bibliography to be of greater use, we needed to expand coverage to include the fisheries, foreign, forestry and systematic zoology literature. To accomplish this, we had to obtain, key-punch, and verify additional citations from university libraries. These additional citations were stored in alphabetical order on a separate magnetic tape. Using the main program MERGE, the additional citations were added to our original file (3000 citations), producing a file of approximately 5000 citations. Professor Schultz arranged for Dowden, Hutchinson, and Ross, Inc. to publish the Bibliography. Ms. Judy Helbling has been instrumental in editorial matters as well as the myriad of other details necessary for publication. In addition, she served capably as a translator of lay English into "publicationese".

GENESIS OF A DATA FILE

FAMULUS allows each user to structure data according to his needs. To effectively use this feature, the user must first decide on a workable set of field names. There are ten fields available and each of these fields, whether needed or not, should be named at the onset. Initially, we used the names DUM1, DUM2, and so on, for those unneeded fields. Later, two of these field names were changed in a run using the program EDIT. In one case, DUM1 was changed to DESC because the field was used for descriptor words (separated by semicolons, as a personal keywording mechanism). Use of the program INDEX on this field allowed an additional indexable field. The use of field names is described more fully in the EDIT section.

Once the fields were named and the data compiled, the information was keypunched. We drew up, with personnel in the keypunch department, a workable set of keypunching instructions (Table 1). They keypunched the data and inserted the necessary blank card between each set of data cards which comprised a citation. Subsequent corrections were made using a local keypunch machine. Our experience indicates that the use of pre-printed forms or cards (Fig. 1) would avoid much of the proofreading necessitated when illegible handwritten input is used and force the collector of each citation to be specific. We have no information on how users at other installations have handled this problem, but believe the form suggested above would enhance efficiency.

Before sending information to keypunch, the user should be aware of the following two factors:

The maximum length of a citation or reference is 4000 characters. This information is found in line 87 of the subroutine SETCOM, page C.109, Appendix C, along with other program limits. (Unless long abstracts are included as a field, we do not believe this restriction will be limiting.)

AUTH ALLEN, K. R.

YEAR 1966.
TITL SOME METHODS FOR ESTIMATING EXPLOITED POPULATIONS.

REF J. FISH. RES. BOARD CAN. 23(10), 1553-1574.

TYPE JOURNAL
CHK POPUL-EST
DESA ALLEN, K.R.

DESC MORTALITY RATE; Q-METHOD; TAG-RECAPTURE ESTIMATION

AUTH ASHBY, E.

YEAR 1936
TITL STATISTICAL ECOLOGY.

REF BOT. REV. 2(5), 221-235

TYPE JOURNAL
CHK FREQ-DIST
DESA ASHBY, E.

DESC POISSON DISTRIBUTION

FIGURE 1. Preprinted Forms for Data

FAMULUS SYSTEY CODING FORM

PAGE ____ OF ____

USER'S NAME M. I. COCHRAN

DATE _____

COMMENTS _____

1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23	24	25	26	27	28	29	30	31	32	33	34	35	36	37	38	39	40	41	42	43	44	45	46	47	48	49	50	51	52	53	54	55	56	57	58	59	60	61	62	63	64	65	66	67	68	69	70	71	72	73	74	75	76	77	78	79	80	
AUTH	ALLEN, K. R.																																																																															
YEAR	1966																																																																															
TITL	SOME METHODS FOR ESTIMATING EXPLOITED POPULATIONS.																																																																															
REF	J. FISH. RES. BOARD CAN. 23(10), 1553-1574.																																																																															
TYPE	JOURNAL																																																																															
CHK	POPUL-EST																																																																															
DESA	ALLEN, K. R.																																																																															
DESC	MORTALITY RATE; Q-METHOD; TAG-RECAPTURE ESTIMATION.																																																																															
AUTH	ALVERSON, D. L. AND G. J. PAULIK																																																																															
YEAR	1973																																																																															
TITL	OBJECTIVES AND PROBLEMS OF MANAGING AQUATIC LIVING RESOURCES.																																																																															
REF	J. FISH. RES. BOARD																																																																															
TYPE	JOURNAL																																																																															
CHK	EXPL-MNG																																																																															
DESA	ALVERSON, D. L.; PAULIK																																																																															
DESC	MODEI																																																																															

FIGURE 2. FAMULUS Coding Form

EDIT

2

1

MAIN PROGRAMS

This section, PROGRAMS, contains explanations for each main program within the FAMULUS set. Explanations for program set-up and information retrieval from the Bibliography are developed here.

EDIT

The EDIT program creates a tape or disc file, makes corrections and adds data submitted from punch cards. Initially, this program was used to submit citations in punch card format to create the first data file. All ten possible fields were named (FIELDS card, Fig. 5), even though DUM1 and DUM2 were not originally used for data. An example setup and flow of the program, system control, and data cards used for a CYBER-70 computer (Fig. 5) are shown in Fig. 4. Explanations for each card in Fig. 4 follow:

(1) SYSTEM JOB CARD

A job is identified by its name (first characters in this card) and certain system resources are requested. We are requesting 72000 Octal words of Central Memory, Time Limit (seconds) and Input-Output (seconds). The system job card must be first in the deck; any other card will be erroneously interpreted by the computer as the job card and cause an error. Only the job name is required.

(2) ACCOUNT

This number was established as a charge code with Boeing Computer Systems Richland, Inc.

(3) MAP

Establishes the type of Memory Load Map desired, with the word 'OFF' in parentheses; the map is not printed.

SYSTEM CONTROL CARDS:

PROGRAM CONTROL CARDS:

SYSTEM CONTROL CARD:

FIGURE 4. System and Program Control Cards for Program EDIT

(4) ATTACH

Makes available (attaches) the file called FAMLIB. FAMULUS was the original name of the program set, but after many revisions it was decided to rename this set FAMLIB. FAMLIB is actually a library-structured system with all the FAMULUS main programs and needed subroutines contained therein.

(5) LIBRARY

Informs the System Loader that FAMLIB is a library of routines and can be searched to satisfy calls for program execution.

(6) REQUEST

If the file is to become permanent, this card must be included. The name in parentheses identifies the device requested; PF means permanent file. The CATALOG (8) card causes this file to be saved.

(7) PROGRAM CALL

Causes execution of program EDIT.

(8) CATALOG

When saving a permanent file, the designations shown in parentheses are followed in order. Previously, we requested that 'TAPE2' be made available as a permanent file and now we also want to store 'TAPE2', renamed SETONE, as a permanent file. The ID (FAMULUS) is a header (title) on SETONE.

(9) END-OF-RECORD CARD

This is a SYSTEM CONTROL CARD which cues the system that the next cards will be the PROGRAM control cards called for during execution of EDIT. The punch on this card is a 7/8/9 multiple punch in column one.

PROGRAM CONTROL CARDS:

(10) ID

When submitting a new set of data to create a data file, EDIT puts the information on this card on the header of the file. In subsequent runs, the ID card must be the first control card. **If it** does not match the ID on the file header, the run immediately stops and the program prints an error message.

(11) DESCRIPTOR

Designates which field is to be used for interrogation by program INDEX.

(12) ORIGINAL

This card is required in the control deck when first creating a file from cards. This informs program EDIT that there is no data file device to access and to immediately begin writing a file using data from cards.

(13) CITATIONS

This card tells EDIT that complete citations in the order indicated by the FIELDS card, follow.

(14) DATA

This is symbolic of all the data cards which would normally follow (as shown in Fig. 5).

(15) END-OF-RECORD CARD

The End of the Program Input (Records).

(16) END-OF-JOB CARD

A System Card which flags the computer that this is the end of the job. This card has a 6/7/8/9 multiple punch in column one.

A list of the control, command, and data cards used as in the above example are shown in Fig. 5 with sample output in Fig. 6.

```

SETONE,CM70000,T100,10200.  MARJI COCHRAN  300/331/2-3251
ACCOUNT(CUB438,CUB438)
MAP(OFF)
ATTACH(FAMLIB,ID=FAMULUS)
REQUEST(TAPE2,*PF)
LIBRARY(FAMLIB)
EDIT.
CATALOG(TAPE2,SETONE,ID=FAMULUS)
X00  ==END OF RECORD CARD=====
/ID/BIBLIOGRAPHY
/FIELDS/(AUTH,YEAR,TITL,REF,TYPE,CHK,DESA,DESC,DUM1,DUM2)
/DESCRIPTOR FIELD/(DESC)
/ORIGINAL/
/CITATIONS/
AUTH ALLEN, K. R.
YEAR 1966.
TITL SOME METHODS FOR ESTIMATING EXPLOITED POPULATIONS.
REF J. FISH. RES. BOARD CAN. 23(10), 1553-1574.
TYPE JOURNAL
CHK POPUL-EST
DESA ALLEN, K. R.
DESC MORTALITY RATE; Q-METHOD; TAG-RECAPTURE ESTIMATION

AUTH ASHBY, E.
YEAR 1936.
TITL STATISTICAL ECOLOGY.
REF BOT. REV. 2(5), 221-235.
TYPE JOURNAL
CHK FREQ-DIST
DESA ASHBY, E.
DESC POISSON DISTRIBUTION

AUTH BARTLETT, M. S.
YEAR 1960.
TITL MONTE CARLO STUDIES IN ECOLOGY AND EPIDEMIOLOGY.
REF PROC. FOURTH BERKELEY SYMP. MATH. STAT. AND PROB. 4, 39-55.
TYPE SYMPOSIUM
CHK STOC-MODEL
DESA BARTLETT, M. S.
DESC COMPUTER

AUTH CASWELL, H.
YEAR 1972
TITL A SIMULATION STUDY OF A TIME LAG POPULATION MODEL.
REF J. THEOR. BIOL. 34(3), 419-439.
TYPE JOURNAL
CHK SIMULAT
DESA CASWELL, H.
DESC VOLTERRA; GAUSE MODEL; COMPUTER

AUTH GULLAND, J. A.
YEAR 1968.
TITL THE CONCEPT OF THE MAXIMUM SUSTAINABLE YIELD AND FISHERY MANAGEMENT,
REF FAO FISHERIES TECHNICAL PAPER NO. 70. I, 13 PP.
TYPE REPORT
CHK EXPL-MNG
DESA GULLAND, J. A.

```

FIGURE 5. Control, Command and Data Cards
for Program EDIT

DESC MODEL

AUTH HASSELL, M. P. AND D. J. ROGERS.

YEAR 1972.

TITL INSECT PARASITE RESPONSES IN THE DEVELOPMENT OF POPULATION MODELS,

REF J. ANIM. ECOL. 41(3), 661-676.

TYPE JOURNAL

CHK POPUL-MODEL

UESA HASSELL, M. P.; ROGERS, D. J.

AUTH JOLLY, G. M.

YEAR 1965.

TITL EXPLICIT ESTIVATES FROM CAPTURE-RECAPTURE DATA WITH BOTH DEATH AND IMMIGRAT
ION-STOCHASTIC MODEL.

REF BIOMETRIKA 52(1/2), 225-247.

TYPE JOURNAL

CHK CAPT-RECAPT

DESA JOLLY, G. M.

AUTH LE CREN, E. D.

YEAR 1973.

TITL SOME EXAMPLES OF THE MECHANISMS THAT CONTROL THE POPULATION DYNAMICS OF SAL
MONID FISH.

REF PP. 125-135. IN, THE MATHEMATICAL THEORY OF THE DYNAMICS OF BIOLOGICAL POPU
LATIONS. (M. S. BARTLETT AND R. W. HIORNS, EDS.), ACADEMIC PRESS* NEW YORK,
NEW YORK.

TYPE BOOK

CHK POPUL-DYNA

DESA LE CREN, E. D.

DESC MODEL; GROWTH RATE

AUTH MAY, R. M.

YEAR 1974.

TITL GENERAL INTRODUCTION.

HEF PP. 1-14. IN, ECOLOGICAL STABILITY. (M. B. USHER AND M. H. WILLIAMSON, EDS.,
), HALSTED PRESS, NEW YORK, NEW YORK.

TYPE BOOK

CHK DIVERSITY

DESA MAY, R. M.

DESC THEORY; HISTORY; YODEL; SPECIES PATTERNS; LOG-NORMAL DISTRIHUTION; EQUITABI
LITY INDICES

AUTH PATTEN, B. C.

YEAR 1963.

TITL INFORMATION PROCESSING BEHAVIOR OF A NATURAL PLANKTON COMMUNITY,

REF AMER. BIOL. TEACH, 25(7), 489-501.

TYPE JOURNAL

CHK DIVERSITY

DESA PATTEN, B. C.

DESC INFORMATION THEORY; ENTROPY

AUTH ROTHSCHILD, B. J.

YEAR 1973.

TITL QUESTIONS OF STRATEGY IN FISHERY MANAGEMENT AND DEVELOPMENT,

REF J. FISH. RES. BOARD CAN, 30(12, PART 2), 2017-2030.

TYPE JOURNAL

CHK EXPL-MNG

FIGURE 5. Control, Command and Data Cards
for Program EDIT (Continued)


```

DESA ROTHSCHILD, B. J.
DESC MODEL

AUTH RYSZKOWSKI, L.
YEAR 1967.
TITL SHORT CUT METHODS FOR THE ESTIMATION OF MEAN LENGTH OF LIFE IN SMALL MAMMAL
      POPULATIONS.
HEF PP. 283-294. IN: SECONDARY PRODUCTIVITY OF TERRESTRIAL ECOSYSTEMS, VOLUME I
  . (K. PETRUSEWICZ, ED.). INSTITUTE OF ECOLOGY, POLISH ACAD. SCI., WARSAW, P
    OLAND.
TYPE BOOK
CHK SURV-MORT
DESA RYSZKOWSKI, L.
DESC SURVIVORSHIP CURVE
X00 ==END OF RECORD CARD=====
W      0/6/7/8/9      MOHAWK  UT-200  END-OF-JOB  CARD
W      0/6/7/8/9      MOHAWK  UT-200  END-OF-JOB  CARD
W      0/6/7/8/9      MOHAWK  UT-200  END-OF-JOB  CARD

```

FIGURE 5. Control, Command and Data Cards
for Program EDIT (continued)

EDIT accepts the following additional program control cards:

/NEW ID/

To change the ID on a file, prepare this card with the new name. **It** comes directly after the original ID card. EDIT changes the ID of the current file upon execution of this card.

/REPLACE/

This feature allows changes in portions of the file. One letter, a word, several words, a phrase, or the entire field can be changed. Four items are required:

- the citation number
- field in which the error is located
- character string in error

the new material

Correction cards should follow the style below. For example, to change the erroneous spelling of AMITH to SMITH in citation #1:

/REPLACE/(1)(AUTH)*AMITH*SMITH*

There are three rules involved in such a correction:

- The citation number and the field label must be enclosed in parentheses.

An asterisk, followed by the data in error, terminated by a second asterisk must enclose the erroneous string.

- The correct information must end with a third asterisk.

EDIT finds the citation number on the existing file, scans the field for the characters enclosed within the first and second asterisks, and replaces them with the next string of characters. Unequal lengths of the two strings are no problem for EDIT. This feature of EDIT can also be

used to delete material within citations. The following card would cause the word **MODEL**, in the **DESC** field in citation #1, to be deleted:

/REPLACE/(1)(DESC)*MODEL**

Following are the rules for deleting material within citations:

- The citation number and the field label are each enclosed in parentheses.
- An asterisk, followed by the data to be deleted, terminated by two asterisks must be used to cause deletion.

Another function of **REPLACE** is to add words or phrases to a field. The following rules are involved:

- The citation number and field label are each enclosed in parentheses.
- Two asterisks, followed by the data to be added, terminated by one asterisk will cause the phrase to be added.

We used the replace card to insert code words as descriptors (c.f. page 3, Background section) for categorizing the Bibliography. For example, use the following card to add the word **BOOK** to the **CHK** field in citation #23.

/REPLACE/(23)(CHK)**BOOK*

If the user needs to add a common word or phrase to several citations in fields which are blank, the use of (1-5) would add the words or phrases to all of the first five citations; subject to the restriction that only 500 citations may be changed in one run.

/DELETE/

This feature allows entire citations to be deleted from the file. This option in EDIT is extremely useful, especially when a citation contains a large number of errors. It is much easier and cheaper to delete an entire citation, correct the original data cards and resubmit the citation into the data file as a brand new citation; subject to the restriction that only 500 citations may be deleted in one run. Citation numbers are enclosed in parentheses following the slash when using this card.

Example:

/DELETE/(2,7,215-221,225)

This instruction would cause the deletion of those citations whose numbers are enclosed in parentheses, (i.e., citation numbers 2, 7, 215 through 221, and 225).

Adding Citations to an Existing File

In order to add citations to an existing data file, EDIT should be used in a procedure called "updating the file." Fig. 7 illustrates how the existing file 'SETONE' was changed by additional card input and the output stored on a new file 'SETTWO'. This second file was alphabetically sorted by author using program SORT, and written as a new file, 'SETNEW'.

Program SORT reads the FIELDS card to ascertain the order the user wants the file arranged. In our example, we used AUTH, YEAR, and TITL as the sort order (see FIELDS card, Fig. 7). The program sorts the AUTH fields in alphabetical order, until it encounters another citation by the same author. It will then check the YEAR fields and sort into chronological order. If two citations by the same author were published in the same year, the program will check the first word of the TITL field to determine the order of the citations. Fields unnamed on the FIELDS card will remain in the order in which they were prior to running the SORT program.

Program control cards have a slash (/) in column 1 before the commands ID, FIELDS, and CITATIONS (Fig. 7), and after the last letter of these **commands**. A detailed flowchart of the steps caused by the cards in Fig. 7 is in Fig. 8. The program prints the ID of the file both before and after sorting (Fig. 9) as well as descriptor and other field names. Three large OKs indicate that processing is complete.

```

INPUT CARD,,./ID/BIBLIOGRAPHY
INPUT CARD,.../FIELDS/(AUTH, YEAR, TITL, REF, TYPE, CHK, DESA, DESC, DUM1, DUM2)
INPUT CARD,.../DESCRIPTOR FIELD/(DESC)
INPUT CARD,,./ORIGINAL/
INPUT CARD,.../CITATIONS/

```

IDENTIFICATION OF OUTPUT TAPE

BIBLIOGRAPHY

```

FIELDS ARE AUTH YEAR TITL REF TYPE CHK DESA DESC DUM1 DUM2
THE DESCRIPTOR FIELD IS DESC
INPUT CARD...AUTH ALLEN, K. R.

```

- 1 AUTH ALLEN, K. R.
YEAR 1966,
TITL SOME METHODS FOR ESTIMATING EXPLOITED POPULATIONS,
REF J. FISH. RES. BOARD CAN. 23(10), 1553-1574,
TYPE JOURNAL
CHK POPUL-EST
DESA ALLEN, K. R.
DESC MORTALITY RATE; Q-METHOD; TAG-RECAPTURE ESTIMATION
- 2 AUTH ASHEY, E.
YEAR 1936.
TITL STATISTICAL ECOLOGY.
REF BOT. REV. 2(5), 221-235,
TYPE JOURNAL
CHK FREQ-DIST
DESA ASHEY, E.
DESC POISSON DISTRIBUTION
- 3 AUTH BARTLETT, M. S.
YEAR 1960,
TITL MONTE CARLO STUDIES IN ECOLOGY AND EPIDEMIOLOGY,
REF PROC. FOURTH BERKELEY SYMP. MATH. STAT. AND PROB. 4, 39-55,
TYPE SYMPOSIUM
CHK STOC-MODEL
DESA BARTLETT, M. S.
DESC COMPUTER
- 4 AUTH CASWELL, H.
YEAR 1972
TITL A SIMULATION STUDY OF A TIME LAG POPULATION MODEL.
REF J. THEOR. BIOL. 34(3), 419-439.
TYPE JOURNAL
CHK SIMULAT
DESA CASWELL, H.
DESC VOLTERRA; GAUSE MODEL; COMPUTER
- 5 AUTH GULLAND, J. A.
YEAR 1968.

FIGURE 6. Output from Program **EDIT**

BIBLIOGRAPHY

06/09/76

- TITL THE CONCEPT OF THE MAXIMUM SUSTAINABLE YIELD AND FISHERY MANAGEMENT.
 REF FAO FISHERIES TECHNICAL PAPER NO. 70, 1, 13 PP.
 TYPE REPORT
 CHK EXPL-MNG
 DESA GULLAND, J. A.
 DESC MODEL
- 6 AUTH HASSELL, M. P. AND D. J. ROGERS.
 YEAR 1972,
 TITL INSECT PARASITE RESPONSES IN THE DEVELOPMENT OF POPULATION MODELS,
 REF J. ANIM. ECOL. 41(3), 661-676.
 TYPE JOURNAL
 CHK POPUL-MODEL
 DESA HASSELL, M. P.; ROGERS, D. J.
- 7 AUTH JOLLY, G. M.
 YEAR 1965,
 TITL EXPLICIT ESTIMATES FROM CAPTURE-RECAPTURE DATA WITH BOTH DEATH AND IMMIGRATION-STOCHASTIC MODEL
 REF BIOMETRIKA 52(1/2), 225-247.
 TYPE JOURNAL
 CHK CAPT-RECAPT
 DESA JOLLY, G. M.
- 8 AUTH LE CREN, E. D.
 YEAR 1973,
 TITL SOME EXAMPLES OF THE MECHANISMS THAT CONTROL THE POPULATION DYNAMICS OF SALMONID FISH.
 REF PP. 125-135, IN: THE MATHEMATICAL THEORY OF THE DYNAMICS OF BIOLOGICAL POPULATIONS. (M. S. BARTLETT AND R. W. HIORNS, EDS.), ACADEMIC PRESS, NEW YORK, NEW YORK,
 TYPE BOOK
 CHK POPUL-DYNA
 DESA LE CREN, E. D.
 DESC MODEL; GROWTH RATE
- 9 AUTH MAY, R. M.
 YEAR 1974,
 TITL GENERAL INTRODUCTION.
 REF PP. 1-14. IN: ECOLOGICAL STABILITY. (M. B. USHER AND M. H. WILLIAMSON, EDS.), HALSTED PRESS, NEW YORK, NEW YORK,
 TYPE BOOK
 CHK DIVERSITY
 DESA MAY, R. M.
 DESC THEORY; HISTORY; MODEL; SPECIES PATTERNS; LOG-NORMAL DISTRIBUTION; EQUITABILITY INDICES
- 10 AUTH PATTEN, B. C.
 YEAR 1963,
 TITL INFORMATION PROCESSING BEHAVIOR OF A NATURAL PLANKTON COMMUNITY,
 REF AMER. BIOL. TEACH. 25(7), 489-501.
 TYPE JOURNAL
 CHK DIVERSITY
 DESA PATTEN, B. C.
 DESC INFORMATION THEORY; ENTROPY
- 11 AUTH ROTHSCHILD, B. J.

FIGURE 6. Output from Program EDIT
(Continued)

```

YEAR 1973.
TITL QUESTIONS OF STRATEGY IN FISHERY MANAGEMENT AND DEVELOPMENT,
REF J. FISH. RES. BOARD CAN. 30(12, PART 2), 2017-2030,
TYPE JOURNAL
CHK EXPL-MNG
OESA ROTHSCHILD, B. J.
DESC MODEL

12 AUTH RYSZKOWSKI, L.
YEAR 1967,
TITL SHORT CUT METHODS FOR THE ESTIMATION OF MEAN LENGTH OF LIFE IN SMALL MAMMAL POPULATIONS.
REF PP. 283-294, IN, SECONDARY PRODUCTIVITY OF TERRESTRIAL ECOSYSTEMS, VOLUME I. (K. PETRUSEWICZ, ED.). INSTITUTE
OF ECOLOGY, POLISH ACAD. SCI., WARSAW, POLAND,
TYPE BOOK
CHK SURV-MORT
DESA RYSZKOWSKI, L.
DESC SURVIVORSHIP CURVE

```

THERE APE 12 CITATIONS ON THE OUTPUT TAPE.
PROCESSING IS COMPLETE.

FIGURE 6. Output from Program **EDIT**
(Continued)


```

SETTWO,CM100000,T100,I0200.  MARJI COCHRAN  300/331/2-3251
ACCOUNT(CUB438,CUB438)
MAP(OFF)
ATTACH(FAML1B,ID=FAMULUS)
ATTACH(SETONE,ID=FAMULUS)
REQUEST(SETNEW,PF)
LIBRARY(FAML1B)
EDIT(,SETONE,SETTWO)
SORT(,SETTWO,SETNEW)
CATALOG(SETNEW,ID=FAMULUS)
X
/ID/BIBLIOGRAPHY
/FIELDS/(AUTH,YEAR,TITL,REF,TYPE,CHK,DESA,DESC,DUM1,DUM2)
/CITATIONS/
AUTH ALVERSON, D. L. AND G. J. PAULIK,
YEAR 1973.
TITL OBJECTIVES AND PROBLEMS OF MANAGING AQUATIC LIVING RESOURCES,
REF J. FISH. RES. BOARD CAN. 30(12, PART 2), 1936-1947,
TYPE JOURNAL
CHK EXPL-MNG
DESA ALVERSON, D. L.;PAULIK, G. J.
DESC MODEL

AUTH BARTLETT, M. S.
YEAR 1957.
TITL ON THEORETICAL MODELS FOR COMPETITIVE AND PREDATORY BIOLOGICAL SYSTEMS,
REF BIOMETRIKA 44(1/2), 27-42.
TYPE JOURNAL
CHK COMPET-DEN
DESA BARTLETT, M. S.

AUTH HEADY, H. F.
YEAR 1955.
TITL TECHNIQUES USEFUL IN RANGE RESEARCH.
REF J. RANGE MANAGE. 8(3), 114-116.
TYPE JOURNAL
CHK SAMPL
DESA HEADY, H. F.
DESC SAMPLING

AUTH INNIS, G.
YEAR 1974.
TITL DYNAMIC ANALYSIS IN "SOFT SCIENCE" STUDIES: IN DEFENSE OF DIFFERENCE EQUATIONS.
REF PP. 102-122. IN, MATHEMATICAL PROBLEMS IN BIOLOGY, LECTURE NOTES IN BIOMATHEMATICS 2. (P. VAN DEN DRIESSCHE, ED.). SPRINGER-VERLAG NEW YORK INC., NEW YORK, NEW YORK.
TYPE BOOK
CHK MATH-MODEL
DESA INNIS, G.
DESC LOTKA-VOLTERRA SYSTEM

AUTH PEARL, R. AND L. J. REED
YEAR 1923.
TITL ON THE MATHEMATICAL THEORY OF POPULATION GROWTH.
REF METRON 3(1), 1-14.
TYPE JOURNAL
CHK POPUL-GRO
DESA PEARL, R.;REED, L. J.
DESC LOGISTIC
XOO **END OF RECORD CARD*****
/ID/BIBLIOGRAPHY
/FIELDS/(AUTH,YEAR,TITL)
XOO **END OF RECORD CARD*****
W 0/6/7/8/9 MOHAWK UT-200 END-OF-JOB CARD
W 0/6/7/8/9 MOHAWK UT-200 END-OF-JOB CARD
W 0/6/7/8/9 MOHAWK UT-200 END-OF-JOB CARD

```

FIGURE 7. Cards Used to Add Additional Citations to an Existing Data File

FIGURE 8. Flowchart of Programs EDIT and SORT as Shown in Figure 7

```

INPUT CARD.../BIBLIOGRAPHY
INPUT CARD.../DESCRIPTOR FIELD/DESA

```

```

IDENTIFICATION OF INPUT TAPE
BIBLIOGRAPHY
FIELDS ARE AUTH YEAR TITL REF TYPE CHK DESA DESC DUM1 DUM2
THE DESCRIPTOR FIELD IS DESC

```

```

IDENTIFICATION OF OUTPUT TAPE
BIBLIOGRAPHY
FIELDS ARE AUTH YEAR TITL REF TYPE CHK DESA DESC DUM1 DUM2
THE DESCRIPTOR FIELD IS DESA

```

```

THERE ARE 17 CITATIONS ON THE OUTPUT TAPE.
PROCESSING IS COMPLETE.

```

```

XXXXX X X
X X X X
X X X X
X X X X
X X X X
X x x x
XXXXX X X

```

```

XXXXX X X
X X X X
X X X X
X X X X
X X X X
X X X X
XXXXX X X

```

```

XXXXX X X
X X X X
X X X X
X X X X
X X X X
X X X X
XXXXX X X

```

FIGURE 9. Output from Programs in Figure 7

GALLEY

GALLEY is the printing program of FAMLIB and will print data files in nearly any desired format. One job card setup, caused the data from one field after another to be printed across the page, is shown in Fig. 10. The ROUTE card is also explained in a footnote to this figure. Fig. 10 shows the control cards used to produce the output shown in Fig. 11. An additional option is to have printing of the citation numbers suppressed (Fig. 12) using the NO NUMBERS option.

The control cards which GALLEY recognizes are:

/ID/

Each main program in FAMLIB first checks the information on this card to see **if it** matches the current ID on the file header. In every run, the ID card must be the first control card. **If it** doesn't match the ID on the file header, the run stops and the program prints an error message.

/SELECT/

This option allows the user to designate specific records to be printed. Any portion or portions of the data file may be printed. For example, the card **/SELECT/(1-70,750,756)** causes the printing of citation numbers 1 through 70, number 750 and 756.

/WIDTH/

This card specifies the printed page width which may range from 20 characters up to the default of 128 characters. Most printers average ten characters per inch so the output page dimensions are from 2 to 13 inches. See Figs. 11 or 12 where we used a width of 56 characters.

/FIELD CHOICE/

Fields to be printed and their output order are selected using this card. Three blank characters are printed between each data field on output. Thus, a new line is not automatically initiated for each output

INPUT CARD.. ./ID/BIBLIOGRAPHY

IDENTIFICATION OF INPUT TAPE

BIBLIOGRAPHY

FIELDS ARE AUTH YEAR TITL REF TYPE CHK DESA DESC DUM1 DUM2

THE DESCRIPTOR FIELD IS DESC

INPUT CARD.../FIELD CHOICE/(AUTH,YEAR,TITL,REF)

INPUT CARD.../SELECT/(1-3,6,14,16)

INPUT CARD.../WIDTH/(56)

BIBLIOGRAPHY

1. ALLEN, K. R. 1966. SOME METHODS FOR ESTIMATING EXPLOITED POPULATIONS. J. FISH. RES. BOARD CAN. 23(10), 1553-1574.
2. ALVEKSON, D. L. AND G. J. PAULIK. 1973. OBJECTIVES AND PROBLEMS OF MANAGING AQUATIC LIVING RESOURCES. J. FISH. RES. BOARD CAN. 30(12, PART 2), 1936-1947.
3. ASHPY, E. 1936. STATISTICAL ECOLOGY. ROT. REV. 2(5), 221-235.
6. CASHWELL, H. 1972. A SIMULATION STUDY OF A TIME LAG POPULATION MODEL. J. THEOR. BIOL. 34(3), 419-439.
14. PATTEN, B. C. 1963. INFORMATION PROCESSING BEHAVIOR OF A NATURAL PLANKTON COMMUNITY. AMER. BIOL. TEACH. 25(7), 489-501.
16. ROTHSCHILD, B. J. 1973. QUESTIONS OF STRATEGY IN FISHERY MANAGEMENT AND DEVELOPMENT. J. FISH. RES. BOARD CAN. 30(12, PART 2), 2017-2030.

BIBLIOGRAPHY

PROCESSING IS COMPLETE

```
XXXXX  X  X
X      X  X  X
X      X  X  X
X      X  X  X
X      X  X  X
X      X  X  X
XXXXX  X  X
```

```
XXXXX  X  X  X
X      X  X  X
X      X  X  X
X      X  X  X
X      X  X  X
X      X  X  X
XXXXX  X  X  X
```

INPUT CARD.../ID/BIBLIOGRAPHY

IDENTIFICATION OF INPUT TAPE

BIBLIOGRAPHY

FIELDS ARE AUTH YEAR TITL REF TYPE CHK UESA DESC DUM1 DUM2

THE DESCRIPTOR FIELD IS DESC

INPUT CARD.../FIELD CHOICE/(AUTH, YEAR, TITL, REF)

INPUT CARD.../SELECT/(1-3,6,14,16)

INPUT CARD.../NO NUMBERS/

INPUT CARD.../WIDTH/(56)

BIBLIOGRAPHY

ALLEN, K. R., 1966. SOME METHODS FOR ESTIMATING
EXPLOITED POPULATIONS. J. FISH. RES. BOARD CAN.
23(10), 1553-1574.

ALVERSON, D. L. AND G. J. PAULIK. 1973. OBJECTIVES
AND PROBLEMS OF MANAGING AQUATIC LIVING RESOURCES, J.
FISH. RES. BOARD CAN. 30(12, PART 2), 1936-1947.

ASHBY, E. 1936. STATISTICAL ECOLOGY. BOT. REV.
2(5), 221-235.

CASWELL, H. 1972. A SIMULATION STUDY OF A TIME LAG
POPULATION MODEL. J. THEOR. BIOL. 34(3), 419-439.

PATTEN, B. C. 1963. INFORMATION PROCESSING BEHAVIOR
OF A NATURAL PLANKTON COMMUNITY, AMER. BIOL. TEACH.
25(7), 489-501.

ROTHSCHILD, B. J. 1973. QUESTIONS OF STRATEGY IN
FISHERY MANAGEMENT AND DEVELOPMENT, J. FISH. RES.
BOARD CAN. 30(12, PART 2), 2017-2030.

BIBLIOGRAPHY

PROCESSING IS COMPLETE

XXXXX	X	X							
X	X	X	X	X					
X	X	X	X	X					
X	X	X	X	X					
X	X	X	X	X					
XXXXX	X	X	X	X					

XXXXX	X	X	X	X					
X	X	X	X	X					
X	X	X	X	X					
X	X	X	X	X					
X	X	X	X	X					
XXXXX	X	X	X	X					

FIGURE 12. Print Out from Program GALLEY
Using Option - No Numbers

NO
 100
 100
 100

INDEX

INDEX compiles an alphabetical list of the words or phrases found in the descriptor field (default in the program) along with a list of the citation numbers in which they are located. Descriptor words or phrases, those words or phrases found in the DESC or DESA fields and separated by semicolon, will be truncated during printing to the program limit of 40 characters.

An example INDEX run is in Fig. 13. The command card INDEX causes execution of that program and produces the output in Fig. 14. In order to change the field indexed from the default descriptor field to the author field (DESA), the cards in Fig. 15 should be used. Fig. 15 also illustrates how EDIT uses the data file SETNEW to create a temporary file (one that we use for just this job) called TEMPY. The next command in the deck executes program INDEX using the file TEMPY and the output is shown in Fig. 16. The file TEMPY is not retained and the original file SETNEW still retains the DESC filed as the designated descriptor field.

INDEX recognizes these control cards:

/ID/

See the Section on program GALLEY.

/SELECT/

See the Section on program GALLEY.

/PUNCH/

This statement is used alone, without citation numbers, to acquire a punched deck of the indexed output.

```

INDEX,CM145000,T100,I050.  MARJI COCHRAN  300/331/2-3251
ACCOUNT(CUB438,CUB438)
MAP(OFF)
ATTACH(SETNEW,ID=FAMULUS)
ATTACH(FAMLIB,ID=FAMULUS)
LIBRARY(FAMLIB)
INDEX(,,,SETNEW)
X00  ==END Of RECORD CARD=====
/ID/BIBLIOGRAPHY
/SELECT/ (1-20)
X00  ==END OF RECORD CARD=====
W      0/6/7/8/9      MOHAWK  UT-200  END-OF-JOB  CARD
W      0/6/7/8/9      MOHAWK  UT-200  END-OF-JOB  CARD

```

FIGURE 13. System and Program Cards Used to Run INDEX

```

INPUT CARD.../ID/BIBLIOGRAPHY
INPUT CARD.../SELECT/(1-20)

```

```

IDENTIFICATION OF INPUT TAPE
BIBLIOGRAPHY
FIELDS ARE  AUTH  YEAR  TITL  REF  TYPE  CHK  DESA  DESC  DUM1  DUM2
THE DESCRIPTOR FIELD IS  DESC

                                STATISTICS
                                CURRENT  MAXIMUM
NUMBER OF DESCRIPTORS                20      4000
NUMBER OF CHARACTERS IN ALL DESCRIPTORS  261     52000
NUMBER OF ENTRIES IN DESCRIPTOR INDEX   25      8000

```

FIGURE 14. Output from Program INDEX

BIBLIOGRAPHY

I N D E X

COMPUTER
5, 6
ENTROPY
14
EQUITABILITY INDICES
13
GAUSE MODEL
b
GROWTH RATE
12
HISTORY
13
INFORMATION THEORY
14
LOG-NORMAL DISTRIBUTION
13
LOGISTIC
15
LOTKA-VOLTERRA SYSTEM
10
MODEL
2, 7, 12, 13, 16
MORTALITY HATE
1
POISSON DISTRIBUTION
3
Q-METHOD
1
SAMPLING
9
SPECIES PATTERNS
13
SURVIVORSHIP CURVE
17
TAG-RECAPTURE ESTIMATION
1
THEORY
13
VOLTERRA
6

F I N I S H E D

FIGURE 14. Output from Program INDEX
(Continued)

```

SETNEW,CM145000,T100,10200.   MARJI COCHRAN   300/331/2
ACCOUNT(CUR438,CUR438)
MAP(OFF)
ATTACH(FAMLIB,ID=FAMULUS)
ATTACH(SETNEW,ID=FAMULUS)
LIBRARY(FAMLIB)
EDIT(,,SETNEW,TEMPY)
INDEX(,,,TEMPY)
X00   ==END OF RECORD CARD=====
/ID/BIBLIOGRAPHY
/DESCRIPTOR FIELD/(DESA)
X00   ==END OF RECORD CARD=====
/ID/BIBLIOGRAPHY
/SELECT/(1-20)
X00   ==END OF RECORD CARD=====
W      0/6/7/8/9      MOHAWK   UT-200   END-OF-JOB   CARD
W      0/6/7/8/9      MOHAWK   UT-200   END-OF-JOB   CARD
W      0/6/7/8/9      MOHAWK   UT-200   END-OF-JOB   CARD

```

FIGURE 15. Control Cards for Use with Programs
EDIT and INDEX Conjointly

```

INPUT CARD,.../ID/BIBLIOGRAPHY
INPUT CARD,.../SELECT/(1-20)

```

```

IDENTIFICATION OF INPUT TAPE
BIBLIOGRAPHY
FIELDS ARE AUTH YEAR TITL REF TYPE CHK DESA DESC DUM1 DUM2
THE DESCRIPTOR FIELD IS DESA

```

	STATISTICS	
	CURRENT	MAXIMUM
NUMBER OF DESCRIPTORS	19	4000
NUMBER OF CHARACTERS IN ALL DESCRIPTORS	237	52000
NUMBER OF ENTRIES IN DESCRIPTOR INDEX	20	8000

FIGURE 16. Output from Control Cards Shown
in Figure 15

BIBLIOGRAPHY

I N D E X

ALLEN, K. R.
1
ALVERSON, D. L.
2
ASHBY, E.
3
BARTLETT, M. S.
4, 5
CASWELL, H.
6
GULLAND, J. A.
7
HASSELL, M. P.
6
HEADY, H. F.
9
INNIS, G.
10
JOLLY, G. M.
11
LE CREN, E. D.
12
MAY, R. M.
13
PATTEN, B. C.
14
PAULIK, G. J.
2
PEARL, R.
15
REED, L. J.
15
ROGERS, D. J.
8
ROTHSCHILD, B. J.
16
RYSZKOWSKI, L.
17

F I N I S H E D

FIGURE 16. Output from Control Cards Shown
in Figure 15. (Continued)

SPORT

_____

1

SORT

SORT arranges the order of records in a data file according to the sequence on the control card. We generally used SORT to sort the data file into an alphabetical order using the order AUTH, YEAR and TITL. Examples of the control cards used and a flow chart are in Fig. 17. However, SORT can reorder any of the fields in any other specified sequence by changes on the Fields card (see EDIT section, Fig. 7).

During construction of the Bibliography we were concerned about consistency of journal abbreviations. Use of the SORT and GALLEY programs gave us the listings needed for aid in proofreading. Control cards which were used to obtain such a listing are shown in Fig. 18. Instead of sorting alphabetically on the AUTH field, we sorted alphabetically on the REF (reference) field. This would print all citations in groups of specific journals, for example, first Amer. Biol. Teach., then Biometrika and etc. as shown in Fig. 19. The citations are listed with the YEAR and AUTH fields to aid us in locating the citations which need corrections. The numbers at the left of each citation (Fig. 19) are not the citation numbers EDIT provides, but a sequential numbering system supplied by SORT. As is shown in Fig. 17 when we sort a file normally, we write the sorted file to a new device and catalog that file. In the case of using the SORT and GALLEY programs for proofreading purposes, we write the new sorted file to a device and do not catalog the file. That is the reason for the designation of TEMPY on the control card for program SORT in Fig. 18.

Another use of this format of using SORT and GALLEY programs to write a temporary (TEMPY) file was in proofreading the DESA field. The control cards for this use of SORT and GALLEY are shown in Fig. 20 with the output in Fig. 21. We chose to print the DESA, YEAR, and REF fields for the same reason as stated above, that the numbers to the left of the citations are not the citation numbers supplied by EDIT, but simply a sequential numbering system supplied by SORT.

/ID/

See Section on program GALLEY.

/FIELDS/

This card specifies the order into which fields are to be rearranged. The labels of the fields to be sorted on, arranged in the desired order, must be enclosed in parentheses. Those fields not specified will remain in their prior order. All ten fields could conceivably be reordered.

/SELECT/

This allows for only sections of the file to be sorted, (i.e., references 500-550), the default is for the entire file to be sorted.

FIGURE 17. Control Cards and Flowchart for Program SORT

```

SETNU2,CM100000,T40,I0200.    MARJI COCHRAN    300/331/2-3251
ACCOUNT(CUB438,CUB438)
MAP(OFF)
ATTACH(SETNU2,ID=FAMULUS)
ATTACH(FAMLIB,ID=FAMULUS)
LIBRARY(FAMLIB)
SORT(,,SETNU2,TEMPY)
GALLEY(,,TEMPY)
X00    ==END OF RECORD CARD=====
/ID/BIBLIOGRAPHY2
/FIELDS/(REF)
X00    ==END OF RECORD CARD=====
/ID/BIBLIOGRAPHY2
/FIELD CHOICE/(REF,YEAR,AUTH)
/SELECT/ (1-25)
/WIDTH/(82)
X00    ==END OF RECORD CARD=====
W      0/6/7/8/9      MOHAWK  UT-200  END-OF-JOB  CARD
W      0/6/7/8/9      MOHAWK  UT-200  END-OF-JOB  CARD

```

FIGURE 18. Control Cards for Programs SORT
and GALLEY

IDENTIFICATION OF INPUT TAPE										
BIBLIOGRAPHY2										
FIELDS ARE	AUTH	YEAR	TITL	REF	TYPE	CHK	DESA	DESC	DUM1	DUM2
THE DESCRIPTOR FIELD IS			DESC							
IDENTIFICATION OF OUTPUT TAPE										
BIBLIOGRAPHY2										
FIELDS ARE	AUTH	YEAR	TITL	REF	TYPE	CHK	DESA	DESC	DUM1	DUM2
ORDERED	REF	AUTH	YEAR	TITL	TYPE	CHK	DESA	DESC	DUM1	DUM2
THE DESCRIPTOR FIELD IS			DESC							

```

      22 CITATIONS IN STRING      1 ON UNIT 2,      22 CITATIONS SO FAR
END OF INTERNAL SORTING
UNIT 2      22 CITATIONS      1 STRINGS
UNIT 4      0 CITATIONS      0 STRINGS

```

END OF PROGRAM.
HAVE SORTED 22 CITATIONS

The figure displays three 10x10 grids, each representing a different stage or configuration of a pattern. The patterns are composed of 'X' characters on a white background.

- Grid 1 (Left):** Shows a square of 'X's with a small cluster at the bottom left. The top row has 6 'X's, and the bottom row has 6 'X's. The left and right sides are also filled with 'X's.
- Grid 2 (Middle):** Shows a similar pattern to Grid 1, but with a different arrangement of 'X's. The top row has 6 'X's, and the bottom row has 6 'X's. The left and right sides are also filled with 'X's.
- Grid 3 (Right):** Shows a more complex pattern with a small cluster at the bottom right. The top row has 6 'X's, and the bottom row has 6 'X's. The left and right sides are also filled with 'X's.

```

INPUT CARD.../ID/BIBLIOGRAPHY2

IDENTIFICATION OF INPUT TAPE
BIBLIOGRAPHY2
FIELDS ARE  AUTH  YEAR  TITL  REF  TYPE  CHK  DESA  CFSC  DUM1  DUM2
ORDERED    REF  AUTH  YEAR  TITL  TYPE  CHK  DESA  DESC  DUM1  DUM2
THE DESCRIPTOR FIELD IS  DESC
INPUT CARD.. ./FIELD CHOICE/(REF,YEAR,AUTH)
INPUT CARD.. ./SELECT/(1-25)
INPUT CARD.../WIDTH/(82)

```

FIGURE 19. Output from Control Cards in Figure 18.

BIBLIOGRAPHY2

1. AMER. BIOL. TEACH. 25(7), 489-501. 1963. PATTEN, B. C.
2. BIOMETRIKA 44(1/2), 27-42. 1957. BARTLETT, M. S.
3. BIOMETRIKA 52(1/2), 225-247. 1965. JOLLY, G. M.
4. HOT. REV. 2(5), 221-235. 1936. ASHBY, E.
5. FAO FISHERIES TECHNICAL PAPER NO. 70. I, 13 PP. 1968. GULLAND, J. A.
6. J. ANIM. ECOL. 41(3), 661-676. 1972. HASSELL, M. P. AND D. J. ROGERS.
7. J. FISH. RES. BOARD CAN. 14(5), 669-681. 1957. SCHAEFER, M. B.
8. J. FISH. RES. BOARD CAN. 23(10), 1553-1574. 1966. ALLEN, K. R.
9. J. FISH. RES. BOARD CAN. 30(12, PART 2), 1936-1947. 1973. ALVERSON, D. L. AND G. J. PAULIK.
10. J. FISH. RES. BOARD CAN. 30(12, PART 2), 2017-2030. 1973. ROTHSCHILD, B. J.
11. J. RANGE MANAGE. 8(3), 114-116. 1955. HEADY, H. F.
12. J. THEOR. BIOL. 34(3), 419-439. 1972. CASWELL, H.
13. METRON 3(1), 1-14. 1923. PEARL, R. AND L. J. REED
14. PP. 1-14. IN, ECOLOGICAL STABILITY. (M. R. USHER AND M. H. WILLIAMSON, EDS.), HALSTED PRESS, NEW YORK, NEW YORK. 1974. MAY, R. M.
15. PP. 102-122. IN, MATHEMATICAL PROBLEMS IN BIOLOGY, LECTURE NOTES IN BIOMATHEMATICS 2. (P. VAN DEN DRIESSCHE, ED.). SPRINGER-VERLAG NEW YORK INC., NEW YORK, NEW YORK. 1974. INNIS, G.
16. PP. 125-135. IN, THE MATHEMATICAL THEORY OF THE DYNAMICS OF BIOLOGICAL POPULATIONS. (M. S. BARTLETT AND R. W. HIORNS, EDS.). ACADEMIC PRESS, NEW YORK, NEW YORK. 1973. LE CREN, E. D.
17. PP. 263-294. IN, SECONDARY PRODUCTIVITY OF TERRESTRIAL ECOSYSTEMS, VOLUME II. (K. PETRUSEWICZ, ED.). INSTITUTE OF ECOLOGY, POLISH ACAD. SCI., WARSAW, POLAND. 1967. RYSZKOWSKI, L.
18. PP. 3-13. IN, PROC. IBM. SCI. COMP. SYMP. STATISTICS, IBM, WHITE PLAINS, NEW YORK. 1965. ROHLF, F. J.
19. PROC. FOURTH BERKELEY SYMP. MATH. STAT. AND PROB. 4, 39-55. 1960. BARTLETT, M. S.
20. PROC. N. Z. ECOL. SOC. 13, 1-4. 1966. SCOTT, D.
21. PROC. N. Z. ECOL. SOC. 16, 58-64. 1969. SCOTT, D.

FIGURE 19. Output from Control Cards in Figure 1%

BIBLIOGRAPHY2

22. PROC. ROYAL SOC. EDINB. SECT. B, BIOL. 137, 433-523, 1950, ZUCKERMAN, S. (LEADERSHIP OF)

PROCESSING IS COMPLETE

The figure consists of three separate diagrams, each showing a grid of points represented by 'X' characters.
 - The left diagram shows a 2D grid. It has a central point 'X'. To its left is a vertical column of five 'X's. To its right is a vertical column of five 'X's. Above the central point is a horizontal row of five 'X's. Below the central point is a horizontal row of five 'X's.
 - The middle diagram shows a 3D grid. It has a central point 'X'. To its left is a vertical column of five 'X's. To its right is a vertical column of five 'X's. Above the central point is a horizontal row of five 'X's. Below the central point is a horizontal row of five 'X's. Additionally, there are points 'X' at the corners of a square formed by the central point and the surrounding points.
 - The right diagram shows a 3D grid. It has a central point 'X'. To its left is a vertical column of five 'X's. To its right is a vertical column of five 'X's. Above the central point is a horizontal row of five 'X's. Below the central point is a horizontal row of five 'X's. Additionally, there are points 'X' at the corners of a square formed by the central point and the surrounding points.

FIGURE 19. Output from Control Cards in Figure 18.
(Continued)


```

SETNU2,CM100000,T40,I0200.  MARJI COCHRAN  300/331/2-3251
ACCOUNT(CUB438,CUB438)
MAP(OFF)
ATTACH(SETNU2,ID=FAMULUS)
ATTACH(FAMLIB,ID=FAMULUS)
LIBRARY(FAMLIB)
SORT(,,SETNU2,TEMPY)
GALLEY(,,TEMPY)
X00  ==END OF RECORD CARD=====
/ID/BIBLIOGRAPHY2
/FIELDS/ (DESA)
X00  ==END OF RECORD CARD=====
/ID/BIBLIOGRAPHY2
/FIELD CHOICE/(DESA,YEAR,REF)
/SELECT/ (1-25)
/WIDTH/(82)
X00  ==END OF RECORD CARD=====
W      0/6/7/8/9      MOHAWK  UT-200  END-OF-JOB  CARD
W      0/6/7/8/9      MOHAWK  UT-200  END-OF-JOB  CARD

```

FIGURE 20. Control Cards to Use Programs SORT and GALLEY to Check for Errors in the DESA Field

INPUT CARD.../ID/BIBLIOGRAPHY2
INPUT CARD.../FIELDS/(DESA)

IDENTIFICATION OF INPUT TAPE

BIBLIOGRAPHY2

FIELDS ARE AUTH YEAR TITL RFF TYPE CHK DESA DESC DUM1 DUM2
THE DESCRIPTOR FIELD IS UESC

IDENTIFICATION OF OUTPUT TAPE

BIBLIOGRAPHY2

FIELDS ARE AUTH YEAR TITL PFF TYPE CHK DESA DESC DUM1 DUM2
ORDERED DESA AUTH YEAR TITL REF TYPE CHK DESC DUM1 DUM2
THE DESCRIPTOR FIELD IS DESC

22 CITATIONS IN STRING 1 ON UNIT 2, 22 CITATIONS SO FAR

END OF INTERNAL SORTING

UNIT 2 22 CITATIONS 1 STRINGS

UNIT 4 0 CITATIONS 0 STRINGS

END OF PROGRAM,

HAVE SORTED 22 CITATIONS

```

XXXXX  X  X
X      X  X  X
X      X  X  X
X      X  X  X
X      X  X  X
X      X  X  X
XXXXX  X  X

```

```

XXXXX  X  X
X      X  X  X
X      X  X  X
X      X  X  X
X      X  X  X
X      X  X  X
XXXXX  X  X

```

```

XXXXX  X  X
X      X  X  X
X      X  X  X
X      X  X  X
X      X  X  X
X      X  X  X
XXXXX  X  X

```

INPUT CARD.../IL/BIBLIOGRAPHY2

IDENTIFICATION OF INPUT TAPE

BIBLIOGRAPHY2

FIELDS ARE AUTH YEAR TITL RFF TYPE CHK DESA DESC DUM1 DUM2
ORDERED DESA AUTH YEAR TITL REF TYPE CHK DESC DUM1 DUM2

THE DESCRIPTOR FIELD IS DESC

INPUT CARD.../FIELD CHOICE/(DESA, YEAR, REF)

INPUT CARD.../SELECT/(1-25)

INPUT CARD.../WIDTH/(82)

FIGURE 21. Output from Control Cards in Figure 20.
(Continued)

BIBLIOGRAPHY2

1. ALLEN, K. R. 1966. J. FISH. RES. BOARD CAN. 23(10), 1553-1574.
2. ALVERSON, D. L.; PAULIK, G. J. 1973. J. FISH. RES. BOARD CAN. 30(12, PART 2), 1936-1947.
3. ASHBY, E. 1936. ROT. REV. 2(5), 221-235.
4. BARTLETT, M. S. 1957. BIOMETRIKA 44(1/2), 27-42.
5. BARTLETT, M. S. 1960. PROC. FOURTH BERKELEY SYMP. MATH. STAT. AND PROB. 4, 39-55.
6. CASWELL, H. 1972. J. THEOR. BIOL. 34(3), 419-439.
7. GULLAND, J. A. 1968. FAO FISHERIES TECHNICAL PAPER NO. 70, 1, 13 PP.
8. HASSELL, M. P.; ROGERS, D. J. 1972. J. ANIM. ECOL. 41(3), 661-676.
9. HEADY, H. F. 1955. J. RANGE MANAGE. 8(3), 114-116.
10. INNIS, G. 1974. PP. 102-122, IN, MATHEMATICAL PROBLEMS IN BIOLOGY, LECTURE NOTES IN BIOMATHEMATICS 2, (P. VAN DEN DRIESSCHE, ED.), SPRINGER-VERLAG NEW YORK INC., NEW YORK, NEW YORK.
11. JOLLY, G. M. 1965. BIOMETRIKA 52(1/2), 225-247.
12. LE CHEN, E. D. 1973. PP. 125-135. IN, THE MATHEMATICAL THEORY OF THE DYNAMICS OF BIOLOGICAL POPULATIONS. (M. S. BARTLETT AND R. W. HIGGINS, EDS.), ACADEMIC PRESS, NEW YORK, NEW YORK.
13. MAY, R. M. 1974. PP. 1-14. IN, ECOLOGICAL STABILITY. (M. B. USHER AND M. H. WILLIAMSON, EDS.), HALSTED PRESS, NEW YORK, NEW YORK.
14. PATTEN, B. C. 1963. AMER. BIOL. TEACH. 25(7), 489-501.
15. PEARL, R.; REED, L. J. 1923. METRON 3(1), 1-14.
16. ROHLF, F. J. 1965. PP. 3-13. IN, PROC. IBM, SCI. COMP. SYMP. STATISTICS, IBM, WHITE PLAINS, NEW YORK.
17. ROTHSCHILD, B. J. 1973. J. FISH. RES. BOARD CAN. 30(12, PART 2), 2017-2030.
18. RYSZKOWSKI, L. 1967. PP. 283-294. IN, SECONDARY PRODUCTIVITY OF TERRESTRIAL ECOSYSTEMS, VOLUME II. (K. PETRUSEWICZ, ED.). INSTITUTE OF ECOLOGY, POLISH ACAD. SCI., WARSAW, POLAND.
19. SCHAEFER, M. B. 1957. J. FISH. RES. BOARD CAN. 14(5), 669-681.
20. SCOTT, D. 1966. PROC. N. Z. ECOL. SOC. 13, 1-4.
21. SCOTT, D. 1969. PROC. N. Z. ECOL. SOC. 16, 58-64.

FIGURE 21. Output from Control Cards in Figure 20.
(Continued)

BIBLIOGRAPHY2

22, ZUCKERMAN, S. 1950. PROC. ROYAL SOC. EDINB. SECT. B, BIOL, 137, 433-523.

PROCESSING IS COMPLETE

```

  XXXXX      X      X
X          X      X      X
X          X      X  X
X          X      X
X          X      X  X
X          X      X      X
  XXXXX      X          X

```

```

  XXXXX      X      X
X        X    X      X
X      X    X X
X    X    X
X    X    X X
X      X    X      X
  XXXXX      X      X

```

```

XXXXX      X      X
X          X      X
X          X      X
X          X      X
X          X      X
XXXXX      X      X

```

FIGURE 21. Output from Control Cards in Figure 20.
(Continued)

MERGE

MERGE

This program combines (merges) a MASTER file and an ADDITIONS file, thus creating a new file.

Initially this was used when we had one data file in a 'near perfect' condition (ie., as many of the errors removed as possible). Additional citations were used to create a new file (ADDITIONS file), as shown in the abbreviated card listing in Fig. 22, and this second file was corrected until it too was in a 'near perfect' condition (Fig. 23). Thus there were now two files which we wished to combine into one.

MERGE only accepts input files (*MASTER* and ~~ADDS~~ having the same number of fields; however, the field names do not have to be the same. The ADDITIONS file, in this section referred to as *SETADD*, was sorted, using program *SORT*, and a new file called *SETNUI* was written and saved. MERGE requires that both files be sorted similarly. Execution of program MERGE allowed the two files (*SETNUI* and *SETNEW*) to be merged and the file *SETNU2* was saved as a new file. The ID on *SETNU2* was changed using a NEW ID card and the data were listed using option *PRINT* in MERGE. An example of the cards required is shown in Fig. 24 and the output is shown in Fig. 25. The output format is the only one allowed using the *PRINT* option.

Of the four files, *SETADD*, *SETNEW*, *SETNUI*, and *SETNU2*, only file *SETNU2* needs to be saved. The others can either be returned or left on the library for some future use.

Control cards which program MERGE recognizes are:

/D/

See Section on program GALLEY.

`/NEW ID/`

See Section on program EDIT.

`/PRINT/`

This statement will print the entire file 128 characters wide. There is no width option.


```

SETADD,CM70000,T50,10100.  MARJI COCHRAN  300/331/2-3251
ACCOUNT(CUR438,CUB438)
ROUTE(OUTPUT,DEF,TID=27,FC=LR),
MAP(OFF)
ATTACH(FAML18,ID=FAMULUS)
REQUEST(TAPE2,*PF)
LIBRARY(FAML18)
EDIT.
CATALOG(TAPE2,SETADD,ID=FAMULUS)
AUDIT(ID=FAMULUS)
X
/ID/BIBLIOGRAPHY
/FIELDS/(AUTH,YEAR,TITL,REF,TYPE,CHK,DESA,DESC,DUM1,DUM2)
/DESCRIPTOR FIELD/(DESC)
/ORIGINAL/
/CITATIONS/
AUTH ROHLF, F. J.
YEAR 1965.
TITL MULTIVARIATE METHODS IN TAXONOMY.
REF PP. 3-13. IN, PROC. IBM. SCI. COMP. SYMP. STATISTICS, IBM, WHITE PLAINS, NE
W YORK,
TYPE BOOK
CHK TAXON
DESA ROHLF, F. J.

AUTH SCHAEFER, M. B.
YEAR 1957.
TITL SOME CONSIDERATIONS OF POPULATION DYNAMICS AND ECONOMICS IN RELATION TO THE
MANAGEMENT OF THE COMMERCIAL MARINE FISHERIES.
REF J. FISH. RES. BOARD CAN. 14(5), 669-681.
TYPE JOURNAL
CHK EXPL-MNG
DESA SCHAEFER, M. B.
DESC MODEL

AUTH SCOTT, D.
YEAR 1966.
TITL INTERPRETATION OF ECOLOGICAL DATA BY PATH ANALYSIS.
REF PROC. N. Z. ECOL. SOC. 13, 1-4.
TYPE JOURNAL
CHK MATH-MODEL
DESA SCOTT, D.

AUTH SCOTT, D.
YEAR 1969.
TITL RELATIONSHIP BETWEEN SOME STATISTICAL METHODS.
REF PROC. N. Z. ECOL. SOC. 16, 58-64.
TYPE JOURNAL
CHK PATTERNS
DESA SCOTT, D.
DESC FACTOR ANALYSIS; S-PRINCIPAL COMPONENT ANALYSIS; CANONICAL ANALYSIS; S-MODEL; R
ANDOMNESS; MULTIVARIATE; ANALYSIS OF VARIANCE; REGRESSION; PATH ANALYSIS; C
OVARIANCE ANALYSIS; DISCRIMINANT ANALYSIS; CORRELATION COEFFICIENT

AUTH ZUCKERMAN S. (LEADERSHIP OF)
YEAR 1950.
TITL A DISCUSSION ON THE MEASUREMENT OF GROWTH AND FORM.
REF PROC. ROYAL SOC. EDINB. SECT. B. BIOL. 137, 433-523.
TYPE JOURNAL
CHK GROW-IND
DESA ZUCKERMAN, S.
X00 ==END OF RECORD CARD=====
W 0/6/7/8/9 MOHAWK UT-200 END-OF-JOB CARD
W 0/6/7/8/9 MOHAWK UT-200 END-OF-JOB CARD

```

FIGURE 22. System and Control Cards Used
for Creating New Data File

```

INPUT CARD.../ID/BIBLIOGRAPHY
INPUT CARD.../FIELDS/(AUTH, YEAR, TITL, REF, TYPE, CHK, DESA, DESC, DUM1, DUM2)
INPUT CARD.../DESCRIPTOR FIELD/(DESC)
INPUT CARD.../ORIGINAL/
INPUT CARD.../CITATIONS/

```

IDENTIFICATION OF OUTPUT TAPE

BIBLIOGRAPHY

FIELDS ARE AUTH YEAR TITL REF TYPE CHK DESA DESC DUM1 DUM2

THE DESCRIPTOR FIELD IS DESC

INPUT CARD.. .AUTH ROHLF, F. J.

- 1 AUTH ROHLF, F. J.
YEAR 1965.
TITL MULTIVARIATE METHODS IN TAXONOMY.
REF PP. 5-13. IN, PROC. IBM, SCI. COMP. SYMP. STATISTICS, IBM, WHITE PLAINS, NEW YORK.
TYPE BOOK
CHK TAXON
DESA ROHLF, F. J.
- 2 AUTH SCHAEFER, M. B.
YEAR 1957.
TITL SOME CONSIDERATIONS OF POPULATION DYNAMICS AND ECONOMICS IN RELATION TO THE MANAGEMENT OF THE COMMERCIAL MARINE FISHERIES.
REF J. FISH. RES. BOARD CAN. 14(5), 669-681.
TYPE JOURNAL
CHK EXPL-MNG
DESA SCHAEFER, M. B.
DESC MODEL
- 3 AUTH SCOTT, D.
YEAR 1966.
TITL INTERPRETATION OF ECOLOGICAL DATA BY PATH ANALYSIS.
REF PROC. N. 2. ECOL. SOC. 13, 1-4.
TYPE JOURNAL
CHK MATH-MODEL
DESA SCOTT, D.
- 4 AUTH SCOTT, D.
YEAR 1969.
TITL RELATIONSHIP BETWEEN SOME STATISTICAL METHODS.
REF PROC. N. 2. ECOL. SOC. 16, 58-64.
TYPE JOURNAL
CHK PATTERNS
DESA SCOTT, D.
DESC FACTOR ANALYSIS; PRINCIPAL COMPONENT ANALYSIS; CANONICAL ANALYSIS; MODEL; RANDOMNESS; MULTIVARIATE; ANALYSIS OF VARIANCE; REGRESSION; PATH ANALYSIS; COVARIANCE ANALYSIS; DISCRIMINANT ANALYSIS; CORRELATION COEFFICIENT
- 5 AUTH ZUCKERMAN, S. (LEADERSHIP OF)
YEAR 1950.
TITL A DISCUSSION ON THE MEASUREMENT OF GROWTH AND FORM.
REF PROC. ROYAL SOC. EDINB. SECT. B. BIOL. 137, 433-523.
TYPE JOURNAL
CHK GROW-IND
DESA ZUCKERMAN, S.

THERE ARE 5 CITATIONS ON THE OUTPUT TAPE.

PROCESSING IS COMPLETE.

```

XXXXX X X
X X X X
X X X X
X X X X
X X X X
XXXXX X X

```

```

XXXXX X X
X X X X
X X X X
X X X X
X X X X
XXXXX X X

```

```

XXXXX X X
X X X X
X X X X
X X X X
X X X X
XXXXX X X

```

FIGURE 23. Output Listing from Figure 22

```

MERGE,CM100000,T200,I0200.  MARJI COCHRAN  300/331/2-3251
ACCOUNT(CUB438,CUB438)
ROUTE(OUTPUT,DEF,TID=27,FC=LR).
MAP(OFF)
ATTACH(SETNEW,ID=FAMULUS)
ATTACH(SETADD,ID=FAMULUS)
REQUEST(SETNU1,*PF)
REQUEST(SETNU2,*PF)
ATTACH(FAMLIB,ID=FAMULUS)
LIBRARY(FAMLIB)
SORT(,,SETADD,SETNU1)
CATALOG(SETNU1,ID=FAMULUS)
MERGE(,,SETNU1,SETNEW,SETNU2)
CATALOG(SETNU2,ID=FAMULUS)
X
/ID/BIBLIOGRAPHY
/FIELDS/(AUTH,YEAR,TITLE)
X
/ID/BIBLIOGRAPHY
/ID/BIBLIOGRAPHY
/NEW ID/BIBLIOGRAPHY2
/PRINT/
X
W          0/6/7/8/9      MOHAWK  UT-200  END-OF-JOB  CARD
W          0/6/7/8/9      MOHAWK  UT-200  END-OF-JOB  CARD

```

FIGURE 24. Control Cards Used To Run Programs
SORT and MERGE

INPUT CARD.../ID/BIBLIOGRAPHY
 INPUT CARD.../FIELDS/(AUTH, YEAR, TITL)

IDENTIFICATION OF INPUT TAPE

BIBLIOGRAPHY

FIELDS ARE AUTH YEAR TITL REF TYPE CHK DESA DESC DUM1 DUM2
 THE DESCRIPTOR FIELD IS DESC

IDENTIFICATION OF OUTPUT TAPE

BIBLIOGRAPHY

FIELDS ARE AUTH YEAR TITL REF TYPE CHK DESA DESC DUM1 DUM2
 THE DESCRIPTOR FIELD IS DESC

5 CITATIONS IN STRING 1 ON UNIT 2, 5 CITATIONS SO FAR

END OF INTERNAL SORTING

UNIT 2 5 CITATIONS 1 STRINGS

UNIT 4 0 CITATIONS 0 STRINGS

END OF PROGRAM.

HAVE SORTED 5 CITATIONS

```

XXXXX  X      X
X      X  X  X
X      X  X  X
X      X  X
X      X  X  X
X      X  X  X
XXXXX  X      X

```

```

XXXXX  X      X
X      X  X  X
X      X  X  X
X      X  X  X
X      X  X  X
X      X  X  X
XXXXX  X      X

```

```

XXXXX  X      X
X      X  X  X
X      X  X  X
X      X  X  X
X      X  X  X
X      X  X  X
XXXXX  X      X

```

INPUT CARD.../ID/BIBLIOGRAPHY
 INPUT CARD.../ID/BIBLIOGRAPHY
 INPUT CARD.../NEW ID/BIBLIOGRAPHY2
 INPUT CARD.../PRINT/

IDENTIFICATION OF ADDITIONS TAPE

BIBLIOGRAPHY

FIELDS ARE AUTH YEAR TITL REF TYPE CHK DESA DESC DUM1 DUM2
 THE DESCRIPTOR FIELD IS DESC

IDENTIFICATION OF MASTER TAPE

BIBLIOGRAPHY

FIELDS ARE AUTH YEAR TITL REF TYPE CHK DESA UESC DUM1 DUM2
 THE DESCRIPTOR FIELD IS DESC

IDENTIFICATION OF OUTPUT TAPE

BIBLIOGRAPHY2

FIELDS ARE AUTH YEAR TITL REF TYPE CHK DESA DESC DUM1 DUM2
 THE DESCRIPTOR FIELD IS DESC

FIGURE 25. Output from Control Cards in Figure 24.

1. ALLEN, K. J. 1966. SOME METHODS FOR ESTIMATING EXPLOITED POPULATIONS. J. FISH. RES. BOARD CAN. 231101. 1953-1974
JOURNAL POPUL-EST ALLEN, K. J. MORTALITY RATE; Q-METHOD; TAG-RECAPTURE ESTIMATION
2. ALVERSON, D. L. AND G. J. PAULIK. 1973. OBJECTIVES AND PROBLEMS OF MANAGING AQUATIC LIVING RESOURCES. J. FISH.
RES. BOARD CAN. 30(12, PART 2), 1936-1947. JOURNAL EXPL-MNG ALVERSON, D. L.; PAULIK, G. J. MODEL
3. ASHBY, E. 1936. STATISTICAL ECOLOGY. BOT. REV. 2(5), 221-235. JOURNAL FREQ-DIST ASHBY, E. POISSON
DISTRIBUTION
4. BARTLETT, M. S. 7. ON THEORETICAL MODELS FOR COMPETITIVE AND PREDATORY BIOLOGICAL SYSTEMS. BIOMETRIKA 44(1/2)
27-42. JOURNAL COMPET-DEN BARTLETT, M. S.
5. BARTLETT, M. S. 1960. MONTE CARLO STUDIES IN ECOLOGY AND EPIDEMIOLOGY. PROC. FOURTH BERKELEY SYMP. MATH. STAT. A
PROB. 4. 39-55. SYMPOSIUM STOC-MODEL BARTLETT, M. S. COMPUTER
6. CASWELL, H. 1972. A SIMULATION STUDY OF A TIME LAG POPULATION MODEL. J. THEOR. BIOL. 34(3), 419-439. JOURNAL
SIMULAT CASWELL, H. VOLTERRA; GAUSE MODEL; COMPUTER
7. GULLAND, J. A. 1968. THE CONCEPT OF THE MAXIMUM SUSTAINABLE YIELD AND FISHERY MANAGEMENT. FAO FISHERIES TECHNICA
PAPER YO. 70, 1, 13 PP. REPORT EXPL-MNG GULLAND, J. A. MODEL
8. HASSELL, M. P. AND D. J. ROGERS. 1972. INSECT PARASITE RESPONSES IN THE DEVELOPMENT OF POPULATION MODELS. J. ANI
ECOL. 41(3), 661-676. JOURNAL POPUL-MODEL HASSELL, M. P.; ROGERS, D. J.
9. HEADY, M. F. 1955. TECHNIQUES USEFUL IN RANGE RESEARCH. J. RANGE MANAGE. 8(3), 114-116. JOURNAL SAMPL
HEADY, M. F. SAMPLING
10. INNIS, G. 1974. DYNAMIC ANALYSIS IN "SOFT SCIENCE" STUDIES: IN DEFENSE OF DIFFERENCE EQUATIONS. PP. 102-122. IN.
MATHEMATICAL PROBLEMS IN BIOLOGY. LECTURE NOTES IN BIOMATHEMATICS 2. (P. VAN DEN DRIESSCHE, ED.). SPRINGER-VERLAG
NEW YORK INC., NEW YORK. BOOK MATH-MODEL INNIS, G. LOTKA-VOLTERRA SYSTEM
11. JOLLY, G. M. 1965. EXPLICIT ESTIMATES FROM CAPTURE-RECAPTURE DATA WITH BOTH DEATH AND IMMIGRATION-STOCHASTIC MODEL
BIOMETRIKA 52(1/2), 225-247. JOURNAL CAPT-RECAPT JOLLY, G. M.
12. LE CREN, E. D. 1973. SOME EXAMPLES OF THE MECHANISMS THAT CONTROL THE POPULATION DYNAMICS OF SALMONID FISH. PP.
125-135. IN. THE MATHEMATICAL THEORY OF THE DYNAMICS OF BIOLOGICAL POPULATIONS. (M. S. BARTLETT AND R. W. HORNES,
EDS.). ACADEMIC PRESS. NEW YORK, NEW YORK. BOOK POPUL-DYNA LE CREN, E. D. MODEL1 GROWTH RATE
13. MAY R. . 1974. GENERAL INTRODUCTION. PP. 1-14. IN. ECOLOGICAL STABILITY. (M. B. USHER AND M. H. WILLIAMSON,
EDS.). HALSTED PRESS, NEW YORK, NEW YORK. BOOK DIVERSITY MAY, R. M. THEORY; HISTORY; MODEL; SPECIES
PATTERNS: LOG-NORMAL DISTRIBUTION; EQUILIBRIUM INDICES
14. PATTEN, B. C. 1963. INFORMATION PROCESSING BEHAVIOR OF A NATURAL PLANKTON COMMUNITY. AMER. BIOL. TEACH. 25(7),
489-501. JOURNAL DIVERSITY PATTEN, B. C. INFORMATION THEORY; ENTROPY
15. PEARL, R. AND L. J. REED 1923. ON THE MATHEMATICAL THEORY OF POPULATION GROWTH. METRON 3(1), 1-14. JOURNAL
POPUL-GRO PEARL, R.; REED, L. J. LOGISTIC
16. ROHLF, F. J. 1965. MULTIVARIATE METHODS IN TAXONOMY. PP. 3-13. IN. PROC. 18M, SCI. COMP. SYMP. STATISTICS, 18M,
WHITE PLAINS, NEW YORK. BOOK TAXON ROHLF, F. J.
17. ROTHSCHILD, B. J. 1973. QUESTIONS OF STRATEGY IN FISHERY MANAGEMENT AND DEVELOPMENT. J. FISH. RES. BOARD CAN.
30(12, PART 2), 2017-2030. JOURNAL EXPL-MNG ROTHSCHILD, B. J. MODEL
18. RYSZKOWSKI, L. 1967. SHORT CUT METHODS FOR THE ESTIMATION OF MEAN LENGTH OF LIFE IN SMALL MAMMAL POPULATIONS. PP
283-294. IN. SECONDARY PRODUCTIVITY OF TERRESTRIAL ECOSYSTEMS. VOLUME I. (K. PETRUSEWICZ, ED.), INSTITUTE OF
ECOLOGY. POLISH ACAD. SCI., WARSAW, POLAND. BOOK SURV-MORT RYSZKOWSKI, L. SURVIVORSHIP CURVE
19. SCHAEFER, M. B. 1957. SOME CONSIDERATIONS OF POPULATION DYNAMICS AND ECONOMICS IN RELATION TO THE MANAGEMENT OF TH
COMMERCIAL MARINE FISHERIES. J. FISH. RES. BOARD CAN. 14(5), 669-681. JOURNAL EXPL-MNG SCHAEFER, M. B.
MODEL
20. SCOTT, D. 1966. INTERPRETATION OF ECOLOGICAL DATA BY PATH ANALYSIS. PROC. N. Z. ECOL. SOC. 13, 1-4. JOURNAL
MATH-MODEL SCOTT, D.
21. SCOTT, D. 1969. RELATIONSHIP BETWEEN SOME STATISTICAL METHODS. PROC. N. Z. ECOL. SOC. 16, 58-64. JOURNAL
PATTERNS SCOTT, D. FACTOR ANALYSIS; PRINCIPAL COMPONENT ANALYSIS; CANONICAL ANALYSIS; MODEL; RANDOMNESS;
MULTIVARIATE; ANALYSIS OF VARIANCE; REGRESSION; PATH ANALYSIS; COVARIANCE ANALYSIS; DISCRIMINANT ANALYSIS;
CORRELATION COEFFICIENT
22. ZUCKERMAN, S. (LEADERSHIP OF) 1950. A DISCUSSION ON THE MEASUREMENT OF GROWTH AND FORM. PROC. ROYAL SOC. EDINB,
SECT. B, BIOL. 137, 433-523. JOURNAL GROW-IND ZUCKERMAN, S.

NUMBER OF CITATIONS ON OUTPUT TAPE ■ 22
PROCESSING COMPLETED

```

XXXXX  X  X
X  X  X  X
X  X  X  X
X  X  X  X
X  X  X  X
X  X  X  X
XXXXX  X  X

XXXXX  X  X
X  X  X  X
X  X  X  X
X  X  X  X
X  X  X  X
X  X  X  X
XXXXX  X  X

XXXXX  X  X
X  X  X  X
X  X  X  X
X  X  X  X
X  X  X  X
X  X  X  X
XXXXX  X  X

```

FIGURE 25. Output from Control Cards in Figure 24 (Continued)

OSSIFY

OSSIFY

OSSIFY will punch cards from any device where a FAMULUS file is stored. Cards will be punched in input format (Fig. 5), with field labels as established by the most current EDIT run. Our principal use of OSSIFY was to produce insurance decks in case the data files on tapes or discs were damaged or accidentally destroyed.

Under certain conditions **it** may be less expensive to use this program for the correction of a badly organized file, particularly **if** the file has long fields, such as abstracts or text-storage which must be deleted or changed extensively. The faulty records could be punched out, and corrected and submitted to the data file as new data using program EDIT. In order to have the data in correct order, the file would then be run on program SORT (Fig. 17).

Control cards needed to run program OSSIFY are shown in Fig. 26, and punched cards produced are in Fig. 27. Note that the citation numbers are not punched in Fig. 27.

OSSIFY recognizes these control cards:

/ID/

See section on program GALLEY.

/SELECT/

This allows the user to punch only certain references, entire sections of the file or the entire file. **If** this card is omitted, the entire file will be punched. **If** the citation numbers are in order, the program will execute faster and be less apt to fail to punch citations.

	USER'S NAME	LOCATION
SETNEW,CM60000,T50.		
ACCOUNT(CUB438,CUB438)		
MAP(OFF)		
*JOBNAME(PUNCH,EAM=1)		
ATTACH(SETNEW,ID=FAMULUS)		
ATTACH(FAMLIB,ID=FAMULUS)		
LIBRARY(FAMLIB)		
OSSIFY(,,,SETNEW)		
ROUTE(PUNCH,TID=C)		
==END-OF-RECORD CARD==		
/ID/ BIBLIOGRAPHY		
/SELECT/ (1,3-6,12,15,19)		
==END-OF-RECORD CARD==		
==END-OF-JOB CARD==		

*The Jobname (System Control Card) is used to identify print and punch files with a job name, current time and date, and the user's name and address from the JOB card (first card in the deck). The job type is called PUNCH and the output cards are to be interpreted, thus the EAM=1 notation. The ROUTE card directs this job to the TID (terminal identification) which in this case is the central site (C).

FIGURE 26. Control Cards Used to Run Program OSSIFY

I MARJI COCHRAN 300 331 2-3251 CUB438 SETNEE4
X00
AUTH ALLEN, K. R.
YEAR 1966.
TITL SOME METHODS FOR ESTIMATING EXPLOITED POPULATIONS.
REF J. FISH. RES. BOARD CAN. 23(10), 1553-1574.
TYPE JOURNAL
CHK POPUL-EST
DESA ALLEN, K. R.
DESC MORTALITY RATE; Q-METHOD; TAG-RECAPTURE ESTIMATION

AUTH ASHBY, E.
YEAR 1936.
TITL STATISTICAL ECOLOGY.
REF BOT. REV. 2(5), 221-235.
TYPE JOURNAL
CHK FREQ-DIST
DESA ASHBY, E.
DESC POISSON DISTRIBUTION

AUTH BARTLETT, M. S.
YEAR 1957.
TITL ON THEORETICAL MODELS FOR COMPETITIVE AND PREDATORY BIOLOGICAL SYSTEMS.
REF BIOMETRIKA 44(1/2), 27-42.
TYPE JOURNAL
CHK COMPET-UEN
DESA BARTLETT, M. S.

AUTH BARTLETT, M. S.
YEAR 1960.
TITL MONTE CARLO STUDIES IN ECOLOGY AND EPIDEMIOLOGY.
REF PROC. FOURTH BERKELEY SYMP. MATH. STAT. AND PROB. 4, 39-55
TYPE SYMPOSIUM
CHK STOC-MODEL
DESA BARTLETT, M. S.
DESC COMPUTER

AUTH CASHWELL, H.
YEAR 1972
TITL A SIMULATION STUDY OF A TIME LAG POPULATION MODEL.
REF J. THEOR. BIOL. 34(3), 419-439.
TYPE JOURNAL
CHK SIMULAT
DESA CASHWELL, H.
DESC VOLTERRAI GAUSE MODEL; COMPUTER

AUTH LE CREN, E. D.
YEAR 1973.
TITL SOME EXAMPLES OF THE MECHANISMS THAT CONTROL THE POPULATION DYNAMICS OF SALMONID FISH.
REF PP. 125-135, IN, THE MATHEMATICAL THEORY OF THE DYNAMICS OF BIOLOGICAL POPULATIONS. (M. S. BARTLETT AND R. W. HIORNS, EDS.), ACADEMIC PRESS, NEW YORK, NEW YORK.
TYPE BOOK
CHK POPUL-DYNA
DESA LE CREN, E. D.
DESC MODEL; GROWTH RATE

AUTH PEARL, R. AND L. J. REED
YEAR 1923.
TITL ON THE MATHEMATICAL THEORY OF POPULATION GROWTH.
REF METRON 3(1), 1-14,
TYPE JOURNAL
CHK POPUL-GRO
DESA PEARL, R.; REED, L. J.
DESC LOGISTIC
X00
X

FIGURE 27. Listing of Punched Cards from Program
OSSIFY in Figure 26

SEARCH

SEARCH

The information retrieval functions of FAMULUS are performed by SEARCH, using any words or phrases desired. The program prints citations which meet the criteria of the search question. The question is a search *formula*, which is constructed by following certain rules.

Any field or combination of fields may be searched. Because of the way we used semicolons as separators in the two fields used for indexing, SEARCH could not be run on these fields. The two fields are DESC (keywords for one of the users) and the DESA (author names in a form for indexing). In order to efficiently execute a search on these two fields, the user would have to know exactly where the semicolons were placed.

The following steps are used when implementing a search: First, define the fields to be scanned. Even though the use of the FIELDS card is a SEARCH program option, for practical purposes **it** is a necessary control card. The program default allows all fields to be searched for each search **formula** which is usually impractical (costly) because strings of interest normally are only in a few fields. Second, prepare a SEARCH card with correct formula using the logical operators shown below:

<u>WORD</u>	<u>PUNCH</u>
<i>and</i>	0-8-7
<i>or</i>	11-0
<i>not</i>	0-8-5

See Table 2 for types of formulas. The terms to be searched are joined by the logical operators shown under "Word" above. The punch codes above were added to the search program by J. R. Tadlock to be compatible with symbols transmitted by our terminal. Since there is no uniformity among keypunches, terminals and/or computer printing units for the symbols representing numeric codes, misspunched logical operators cannot be located by visual inspection interpreted cards or card listings.

TABLE 2. Logic of Search Requests*

<u>Formula</u>	<u>Results</u>
<i>A</i>	<i>Prints citations in which A appears.</i>
<i>A&_B</i>	<i>Prints citations in which A appears, but not B.</i>
<i>A&_B&_C</i>	<i>Prints citations in which A appears, but not B and not C. This can be extended to include as many words or phrases as desired.</i>
<i>A & B</i>	<i>Prints citations in which both A and B appear.</i>
<i>A & B & C & D</i>	<i>Prints citations in which A, B, C, and D appear. This can be extended to include as many words or phrases as desired.</i>
<i>A ! B</i>	<i>Prints citations in which A or B appear.</i>
<i>A & B ! C & D</i>	<i>Prints citations in which A and B appear, or C and D.</i>
<i>A & B &_C ! D</i>	<i>Prints citations in which A and B appear, but not C or D.</i>

---

*U.S. Dept. Agriculture, 1969.

In Fig. 28 the ! symbol represents the character for 'or', and & represents the character for 'and', and the _ (underline) symbol represents the 'not' character in a SEARCH formula.

The first search request is for FISH in the TITL field and for the word REPORT in the TYPE field. The second request is for MARINE or AQUATIC and not FISH in the TITL field. The third formula searches for MARINE or AQUATIC(and FISH) combined with PLANKTON or BENTHOS(and POPULATION). The fourth formula requests MARINE or AQUATIC or FISH or POPULATION and FISH in the REF field. The fifth formula will print all the citations which have been categorized under DIVERSITY in the CHK field. The last formula obtains all citations categorized as BOOK in the TYPE field. The results of this job are printed in Fig. 29.

```

SETN,CM60000,T200,I0500.          USER'S NAME          LOCATION
ACCOUNT(CUB438,CUB438)
MAP(OFF)
ATTACH(SETNU2,ID=FAMULUS)
ATTACH(FAMLIB,ID=FAMULUS)
LIBRARY(FAMLIB)
SEARCH( ,SETNU2)
==END-OF-RECORD-CARD==
/ID/BIBLIOGRAPHY2
/FIELD CHOICE/(AUTH,YEAR,TITL,REF,DESC,CHK)
/WIDTH/ (56)
/SELECT/ (1-25)
/FIELDS/(TITL)
/SEARCH/FISH&REPORT(TYPE)
/SEARCH/MARINE!AQUATIC&_FISH
/SEARCH/MARINE!AQUATIC&FISH!PLANKTON!BENTHOS&POPULATION
/SEARCH/MARINE!AQUATIC!FISH! POPULATION&FISH(REF)
/SEARCH/DIVERSITY(CHK)
/SEARCH/BOOK(TYPE)
==END-OF-RECORD-CARD==
==END-OF-JOB-CARD==

```

FIGURE 28. Control Cards Used To Run Program SEARCH

- Use of the '*and*' operator implies the user wants to retrieve only citations in which **all** of the words or phrases joined by *and* are present. If a single word or phrase is missing, the citation will not be selected.
- The '*or*' operator indicates the user wants the citations in which the first word or phrase **or** the second *or* etc. are present. Any or all of the requested words or phrases could be in the citation which program SEARCH prints.
- The '*not*' operator specifies that citations containing the word or phrase preceded by *not* are not to be printed.

Truncated words can be used to retrieve those citations which contain longer versions, e.g., *ECOLOG* would retrieve citations in which *ECOLOGY* or *ECOLOGICAL* would occur.

Multiple searches can be executed by following the first search formula with a second, third, etc. There is no programmed maximum limitation on the number of searches allowed in one run, but costs are proportional to the number of searches. See Fig. 30 for print out from several SEARCH requests.

As was stated above, the requested words or phrases are located only in the fields specified by the FIELDS card. However, sometimes there is a need to include a word or phrase from another field, e.g., a requester may wish those citations in which MODEL or MODELING occurs, but wants only those written by *SMYTHE* and only for the years 1950-1959. Such a search formula would be written as follows:

/SEARCH/ (MODEL) and SMYTHE(AUTH) and 1950(YEAR)

Another option allowed by the program (untested by us) is writing the retrieved set of citations onto a disc file or tape device instead of output to the printer. Output is allowed in two formats, either a non-field-labeled format or a card format. The card format allows field labels and

thus the file is similar to one created from punch cards. The only disadvantage to use of the option is that citations retrieved from only one search formula can be written onto a tape, so several independent search formulas would require extraordinary resources.

SEARCH recognizes the following program control cards:

/ID/

See section on program GALLEY.

/SELECT/

See section on program GALLEY.

/NUMBERS/

For some purposes the only output desired will be citation numbers. This control card limits the output to printing of just the citation numbers, not the content of the citation, see Fig. 12 for an example of output.

/WIDTH/

See section on program GALLEY.

/FIELD CHOICE/

See section on program GALLEY.

INPUT CARD.../ID/BIBLIOGRAPHY2
 IDENTIFICATION OF INPUT TAPF
 BIBLIOGRAPHY2
 FIELDS ARE AUTH YEAR TITL REF TYPE CHK DESA DESC DUM1 DUM2
 THE DESCRIPTOR FIELD IS DESC

INPUT CARD.../FIELD CHOICE/(AUTH, YEAR, TITL, REF, DESC, CHK)
 INPUT CARD.../WIDTH/(56)
 INPUT CARD.../SELECT/(1-25)
 INPUT CARD.../FIELDS/(TITL)
 INPUT CARD.../SEARCH/FISH&REPORT(TYPE)

7. GULLAID, J. A. 1968. THE CONCEPT OF THE
 MAXIMUM SUSTAINABLE YIELD AND FISHERY MANAGEMENT.
 FAO FISHERIES TECHNICAL PAPER NO. 70. 1, 13 PP.
 MODEL EXPL-MNG

1 CITATIONS SATISFY THE SEARCH REQUEST

INPUT CARD.../SEARCH/MARINE!AQUATIC&FISH

2. ALVERSON, D. L. AND G. J. PAULIK. 1973.
 OBJECTIVES AND PROBLEMS OF MANAGING AQUATIC
 LIVING RESOURCES. J. FISH. RES. BOARD CAN.
 30(12, PART 2), 1936-1947. MODEL EXPL-MNG
19. SCHAEFER, M. H. 1957. SOME CONSIDERATIONS OF
 POPULATION DYNAMICS AND ECONOMICS IN RELATION TO
 THE MANAGEMENT OF THE COMMERCIAL MARINE
 FISHERIES. J. FISH. RES. BOARD CAN. 14(5),
 669-681. MODEL EXPL-MNG

2 CITATIONS SATISFY THE SEARCH REQUEST.

INPUT CARD.../SEARCH/MARINE!AQUATIC&FISH!PLANKTON!BENTHOS&POPULATION

14. PATTEN, B. C. 1971. INFORMATION PROCESSING
 BEHAVIOR OF 4 NATURAL PLANKTON COMMUNITY. AMER
 BIOL. TEACH. 25(7), 489-501. INFORMATION
 THEORY; ENTROPY DIVERSITY
19. SCHAEFER, M. H. 1957. SOME CONSIDERATIONS OF
 POPULATION DYNAMICS AND ECONOMICS IN RELATION TO
 THE MANAGEMENT OF THE COMMERCIAL MARINE
 FISHERIES. J. FISH. RES. BOARD CAN. 14(5),
 669-681. MODEL EXPL-MNG

FIGURE 29. Output of Control Cards in Figure 28.

BIBLIOGRAPHY2

2 CITATIONS SATISFY THE SEARCH REQUEST.

INPUT CARD.../SEARCH/MARINE!AQUATIC!FISH!POPULATION&FISH(REF)

1. ALLEN, K. R. 1966. SOME METHODS FOR ESTIMATING EXPLOITED POPULATIONS. J. FISH. RES. BOARD CAN. 23(10), 1553-1574. MORTALITY RATE; Q-METHOD; TAG-RECAPTURE ESTIMATION POPUL-EST
2. ALVERSON, D. L. AND G. J. PAULIK. 1973. OBJECTIVES AND PROBLEMS OF MANAGING AQUATIC LIVING RESOURCES. J. FISH. RES. BOARD CAN. 30(12, PART 2), 1936-1947. MODEL EXPL-MNG
7. GULLAND, J. A. 1969. THE CONCEPT OF THE MAXIMUM SUSTAINABLE YIELD AND FISHERY MANAGEMENT. FAO FISHERIES TECHNICAL PAPER NO. 70, 1, 13 PP. MODEL EXPL-MNG
12. LE CREN, E. D. 1973. SOME EXAMPLES OF THE MECHANISMS THAT CONTROL THE POPULATION DYNAMICS OF SALMONID FISH. PP. 125-135. IN, THE MATHEMATICAL THEORY OF THE DYNAMICS OF BIOLOGICAL POPULATIONS. (M. S. BARTLETT AND R. W. HIGGINS, EDS.), ACADEMIC PRESS, NEW YORK, NEW YORK. MODEL; GROWTH RATE POPUL-DYNA
17. RUTHSCHILD, R. J. 1973. QUESTIONS OF STRATEGY IN FISHERY MANAGEMENT AND DEVELOPMENT. J. FISH. RES. BOARD CAN. 30(12, PART 2), 2017-2030. MODEL EXPL-MNG
19. SCHAEFER, M. B. 1957. SOME CONSIDERATIONS OF POPULATION DYNAMICS AND ECONOMICS IN RELATION TO THE MANAGEMENT OF THE COMMERCIAL MARINE FISHERIES. J. FISH. RES. BOARD CAN. 14(5), 669-681. MODEL EXPL-MNG

6 CITATIONS SATISFY THE SEARCH REQUEST.

INPUT CARD.../SEARCH/DIVERSITY(CHK)

13. MAY, H. M. 1974. GENERAL INTRODUCTION. PP. 1-14. IN, ECOLOGICAL STABILITY. (M. B. USHER AND M. H. WILLIAMSON, EDS.), HALSTED PRESS, NEW YORK, NEW YORK. THEORY; HISTORY; MODEL; SPECIES PATTERNS; LOG-NORMAL DISTRIBUTION; EQUITABILITY INDICES DIVERSITY
14. PATTEN, B. C. 1963. INFORMATION PROCESSING BEHAVIOR OF A NATURAL PLANKTON COMMUNITY. AMER BIOL. TEACH. 25(7), 489-501. INFORMATION THEORY; ENTROPY DIVERSITY

2 CITATIONS SATISFY THE SEARCH REQUEST.

FIGURE 29. Output of Control Cards in Figure 28.
(Continued)

INPUT CARD.. ./SEARCH/HOOK(TYPE)

10. INNIS, G., 1974. DYNAMIC ANALYSIS IN "SOFT SCIENCE" STUDIES: IN DEFENSE OF DIFFERENCE EQUATIONS. PP. 102-122. IN, MATHEMATICAL PROBLEMS IN BIOLOGY. LECTURE NOTES IN BIOMATHEMATICS 2, (P. VAN DEN DRIESSCHE, ED.), SPRINGER-VERLAG NEW YORK INC., NEW YORK, NEW YORK. LOTKA-VOLTERRA SYSTEM MATH-MODEL
12. LE CREN, E. D., 1973. SOME EXAMPLES OF THE MECHANISMS THAT CONTROL THE POPULATION DYNAMICS OF SALMONID FISH. PP. 125-135. IN, THE MATHEMATICAL THEORY OF THE DYNAMICS OF BIOLOGICAL POPULATIONS, (M. S. BARTLETT AND R. W. HIGGINS, EDS.), ACADEMIC PRESS, NEW YORK, NEW YORK. MODEL; GROWTH RATE POPUL-UYNA
13. MAY, R. M., 1974. GENERAL INTRODUCTION. PP. 1-14. IN, ECOLOGICAL STABILITY. (M. B. USHER AND M. H. WILLIAMSON, EDS.), HALSTED PRESS, NEW YORK, NEW YORK. THEORY; HISTORY; MODEL; SPECIES PATTERNS; LOG-NORMAL DISTRIBUTION; EQUITABILITY INDICES DIVERSITY
16. RUHLF, F. J., 1965. MULTIVARIATE METHODS IN TAXONOMY. PP. 3-13. IN, PROC. IBM, SCI. COMP. SYMP. STATISTICS* IBM, WHITE PLAINS, NEW YORK. TAXON
18. RYSZKOWSKI, L., 1967. SHORT CUT METHODS FOR THE ESTIMATION OF MEAN LENGTH OF LIFE IN SMALL MAMMAL POPULATIONS. PP. 283-294. IN, SECONDARY PRODUCTIVITY OF TERRESTRIAL ECOSYSTEMS, VOLUME II. (K. PETRUSEWICZ, ED.), INSTITUTE OF ECOLOGY, POLISH ACAD. SCI., WARSAW, POLAND. SURVIVORSHIP CURVE SURV-MORT

5 CITATIONS SATISFY THE SEARCH REQUEST.

END OF SEARCHING.

```

XXXXXX  X  X
X  X  X  X  X
X  X  X  X  X
X  X  X  X  X
X  X  X  X  X
XXXXXX  X  X

XXXXXX  X  X
X  X  X  X  X
X  X  X  X  X
X  X  X  X  X
X  X  X  X  X
XXXXXX  X  X

XXXXXX  X  X
X  X  X  X  X
X  X  X  X  X
X  X  X  X  X
X  X  X  X  X
XXXXXX  X  X

```

FIGURE 29. Output of Control Cards in Figure 28.
(Continued)

VOCAB

Program VOCAB uses a designated **field(s)** to compile an alphabetical list of all words in the chosen **field(s)**. This program can be used to obtain an alphabetical list of words found in several fields. However, correcting errors would be more difficult because the user would not know which field in a citation contained the word in error. A vocabulary list from the TITL field is an asset when constructing a logical formula for use with program SEARCH. This list can also be used as an index of major headings to categorize the data file. Also proofreading for errors in the title field is easier using program VOCAB.

This program uses the /ID/, /FIELDS/, and /SELECT/ cards as do the other programs. The /FIELDS/, however, is used to designate fields which the user wants compiled into an alphabetical list, rather than fields to be printed (Fig. 30). A list is compiled of prepositions and articles such as: the, and, between, and to. These words are considered as "stop words" or "trivial words." The /VOCABULARY/ card contains these trivial words separated by asterisks followed by "starter" words, which are words known to be in the designated field. Starter words are separated by commas. The User's Manual indicates one starter word is sufficient, but we have always included three. An example of a /VOCABULARY/ card would be:

```
/VOCABULARY/A*AND*BETWEEN*BOTH*BY*FOR*IN*OF*ON*MARINE,STUDY,POPULATION
```

VOCAB first prints the 'input cards', identification of tape (Fig. 31) and then prints a list of the trivial words and their respective citation numbers (Fig. 32). On the following pages, three columns of words found in the selected **field(s)**, are printed. These are accompanied by the number of times each word appears and citation numbers for the first three occurrences are in Fig. 33.

	USER'S NAME	LOCATION
SETN2,CM202000,T100,I0500.		
ACCOUNT(CUB438,CUB438)		
MAP(OFF)		
ATTACH(SETNU2,ID=FAMULUS)		
ATTACH(FAMLIB,ID=FAMULUS)		
LIBRARY(FAMLIB)		
VOCAB(,,,SETNU2)		
==END-OF-RECORD-CARD==		
/ID/BIBLIOGRAPHY2		
/FIELDS/ (TITL)		
/SELECT/ (1-25)		
/VOCABULARY/A*AND*BETWEEN*BOTH*BY*FOR*FROM*IN*OF*ON*SOME*THAT*THE*TO*		
WITH*BOTH*MARINE,POPULATION,STUDY		
==END-OF-RECORD-CARD==		
==END-OF-JOB-CARD==		

FIGURE 30. Control Cards Used To Run Program VOCAB

```

INPUT CARD.../ID/BIBLIOGRAPHY2
INPUT CARD.../FIELDS/(TITL)
INPUT CARD.../SELECT/(1-25)
INPUT CARD.../VOCABULARY/A*AND*BETWEEN*BOTH*BY*FOR*FROM*IN*OF*ON*SOME*THAT*THE*TO*WITH*BOTH*M
INPUT CARD...ARINE,POPULATION,STUDY

```

```

IDENTIFICATION OF INPUT TAPE
BIBLIOGRAPHY2
FIELDS ARE  AUTH  YEAR  TITL  REF  TYPE  CHK  DESA  DESC  DUM1  DUM2
THE DESCRIPTOR FIELD IS  DESC

```

FIGURE 31. First Page of Output from VOCAB

BIBLIOGRAPHY2

T R I V I A L W O R D S				
A				
AND	4(	6	6	14)
BETWEEN	8(	2	4	5)
BOTH	1(	21		)
BY	1(	11		)
FOR	1(	20		)
FROM	3(	1	4	18)
IN	1(	11		)
OF	9(	5	8	9)
ON	16(	2	6	7)
SOME	3(	4	15	22)
THAT	4(	1	12	19)
THE	1(	12		)
TO	10(	7	7	8)
WITH	1(	19		)
	1(	11		)

FIGURE 32. Output from VOCAB: Trivial Words, Number of Occurrences and First Three Citation Numbers

BIBLIOGRAPHY2

BIBLIOGRAPHY2				V O C A B U L A R Y									
WSOFT	1(	10	)	EPIDEMIOLOGY	1(	5	)	MANAGING	1(	2	)		
ANALYSIS	2(	10	20	)	EQUATIONS	1(	10	)	MARINE	1(	17	)	
AQUATIC	1(	2	)	ESTIMATES	1(	11	)	MATHEMATICAL	1(	15	)		
BEHAVIOR	1(	14	)	ESTIMATING	1(	1	)	MAXIMUM	1(	7	)		
BIOLOGICAL	1(	4	)	ESTIMATION	1(	18	)	MEAN	1(	18	)		
CAPTURE-RECAPTURE	1(	11	)	EXAMPLES	1(	12	)	MEASUREMENT	1(	22	)		
CARLO	1(	5	)	EXPLICIT	1(	11	)	MECHANISMS	1(	12	)		
COMMERCIAL	1(	19	)	EXPLOITED	1(	1	)	METHODS	4(	1	16	16)	
COMMUNITY	1(	14	)	FISH	1(	12	)	MODEL	2(	5	11	)	
COMPETITIVE	1(	4	)	FISHERIES	1(	19	)	MODELS	2(	4	5	)	
CONCEPT	1(	7	)	FISHERY	2(	7	17	)	MONTE	1(	5	)	
CONSIDERATIONS	1(	19	)	FORM	1(	22	)	MULTIVARIATE	1(	15	)		
CONTROL	1(	12	)	GENERAL	1(	13	)	NATURAL	1(	14	)		
CUT	1(	18	)	GROWTH	2(	15	22	)	OBJECTIVES	1(	2	)	
DATA	2(	11	20	)	IMMIGRATION-STOCHASTIC	1(	11	)	PARASITE	1(	5	)	
DEATH	1(	11	)	INFORMATION	1(	14	)	PATH	1(	20	)		
DEFENSE	1(	10	)	INSECT	1(	8	)	PLANKTON	1(	14	)		
DEVELOPMENT	2(	8	17	)	INTERPRETATION	1(	20	)	POPULATION	5(	6	5	12)
DIFFERENCE	1(	10	)	INTRODUCTION	1(	13	)	POPULATIONS	2(	1	13	)	
DISCUSSION	1(	22	)	LAG	1(	6	)	PREDATORY	1(	4	)		
DYNAMIC	1(	10	)	LENGTH	1(	16	)	PROBLEMS	1(	2	)		
DYNAMICS	2(	12	19	)	LIFE	1(	18	)	PROCESSING	1(	14	)	
ECOLOGICAL	1(	20	)	LIVING	1(	2	)	QUESTIONS	1(	17	)		
ECOLOGY	2(	5	)	MAMMAL	1(	18	)	RANGE	1(	7	)		
ECONOMICS	1(	19	)	MANAGEMENT	3(	7	17	19)	RELATION	1(	19	)	

FIGURE 33. Output from VOCAB: Vocabulary List, Number of Occurrences and the First Three Citation Number

BIBLIOGRAPHY2

V O C A B U L A R Y

RELATIONSHIP	1 (	21	)
RESEARCH	1 (	9	)
RESOURCES	1 (	2	)
RESPONSES	1 (	8	)
SALMONID	1 (	12	)
SCIENCE"	1 (	10	)
SHORT	1 (	18	)
SIMULATION	1 (	6	)
SMALL	1 (	18	)
STATISTICAL	2 (	3	21
STRATEGY	1 (	17	)
STUDIES	2 (	5	10
STUDY	1 (	6	)
SUSTAINABLE	1 (	7	)
SYSTEMS	1 (	4	)
TAXONOMY	1 (	16	)
TECHNIQUES	1 (	9	)
TWEORETICAL	1 (	4	)
THEORY	1 (	15	)
TIME	1 (	6	)
USEFUL	1 (	9	)
YIELD	1 (	7	)

FIGURE 33. Output from VOCAB: Vocabulary List, Number of Occurrences and the First Three Citation Numbers. (Continued)

VOCAB Statistics

After all the words are printed in columns, a set of "statistics" are printed (Fig. 34). These are a user aid to understand the composition of the file in terms of numbers of meaningless (trivial) words, their length, the number of vocabulary words, and the size of the selected **field(s)** by numbers of distinct words. Finally, "the last citation number inspected" notation informs the user how far VOCAB has progressed into the data file (Fig. 34). The rest of the statistics are self-explanatory.

Punched Output

The optional punched output consists of a punched deck of stop words, plus the words in the TRIVIAL WORD and VOCABULARY lists. This deck can be used as input to the next program run. A list of the cards which were produced when a /PUNCH/ card was used is shown in Fig. 35.

VOCAB recognizes these control cards:

/ID/

Matches the ID on the file attached.

/FIELDS/

Unlike the other programs this card is used by VOCAB to designate which field or fields are to be scanned to compile the alphabetic listings of TRIVIAL and VOCABULARY words. The default, or field to be used **if** this card is not used, is the field designated by the last program EDIT run as the descriptor field.

/SELECT/

Allows the user to designate only certain sections of the file, for instance, the first 10 citations or the middle section, etc. This card is optional; the default is the entire file.

B IOGRAPH

THE LAST CITATION INSPECTED WAS 22

STATISTICS FOR OUTPUT DICTIONARY

TRIVIAL WORDS...

DISTRIBUTION OF WORD LENGTHS

1	5	3	5	0	0	1

NUMBER OF TRIVIAL WORDS = 15
 AVERAGE LENGTH = 3.1
 MAXIMUM LENGTH = 7
 MAXIMUM POSSIBLE = 40

VOCABULARY WORDS...

DISTRIBUTION OF WORD LENGTHS

0	0	2	7	50	9	13	15	10	14
8	5	0	2	0	0	1	0	0	0
0	1								

NUMBER OF VOCABULARY ENTRIES = 97
 AVERAGE LENGTH = 8.2
 MAXIMUM LENGTH = 22
 MAXIMUM POSSIBLE = 40

DICTIONARY SIZE...

NUMBER OF ENTRIES = 112	MAXIMUM = 6500
NUMBER OF CHARACTERS = 841	MAXIMUM = 54000

```

xxxxx  x  x
x      x  x  x
x      x  x  x
x      x  x  x
x      x  x  x
xxxxx  x  x

```

```

xxxxx  x  x
x      x  x  x
x      x  x  x
x      x  x  x
x      x  x  x
xxxxx  x  x

```

```

xxxxx  x  x
x      x  x  x
x      x  x  x
x      x  x  x
x      x  x  x
xxxxx  x  x

```

FIGURE 34. Statistics Obtained as Output from Program VOCAB

```

EZ
I MARJI COCHRAN 300 331 2-3251 CUB438 SETNUEZ :7 16.37.48 07/23/76
X00
A*AND*BETWEEN*BOTH*BY*FOR*FROM*IN*OF*ON*SOME*THAT*THE*TO*WITH*"SOFT,ANALYSIS,AQU
ATIC,BEHAVIOR,BIOLOGICAL,CAPTURE-RECAPTURE,CARLO,COMMERCIAL,COMMUNITY,COMPETITIV
E,CONCEPT,CONSIDERATIONS,CONTROL,CUT,DATA,DEATH,DEFENSE,DEVELOPMENT,DIFFERENCE,D
ISCUSSION,DYNAMIC,DYNAMICS,ECOLOGICAL,ECOLOGY,ECONOMICS,EPIDEMIOLOGY,EQUATIONS,E
STIMATES,ESTIMATING,ESTIMATION,EXAMPLES,EXPLICIT,EXPLOITED,FISH,FISHERIES,FISHER
Y,FORM,GENERAL,GROWTH,IMMIGRATION-STOCHASTIC,INFORMATION,INSECT,INTERPRETATION,I
NTRODUCTION,LAG,LENGTH,LIFE,LIVING,MAMMAL,MANAGEMENT,MANAGING,MARINE,MATHEMATICA
L,MAXIMUM,MEAN,MEASUREMENT,MECHANISMS,METHODS,MODEL,MODELS,MONTE,MULTIVARIATE,NA
TURAL,OBJECTIVES,PARASITE,PATH,PLANKTON,POPULATION,POPULATIONS,PREDATORY,PROBLEM
S,PROCESSING,QUESTIONS,RANGE,RELATION,RELATIONSHIP,RESEARCH,RESOURCES,RESPONSES,
SALMONID,SCIENCE",SHORT,SIMULATION,SMALL,STATISTICAL,STRATEGY,STUDIES,STUDY,SUST
AINABLE,SYSTEMS,TAXONOMY,TECHNIQUES,THEORETICAL,THEORY,TIME,USEFUL,YIELD
X00
X

```

FIGURE 35 Punched Card Output from Program VOCAB.

/VOCABULARY/

This card contains the stop words and one or more terminal starter words, separated by **commas**.

Example:

**/VOCABULARY/ *AND*BUT*CAN*HAD*OF*ON*THE*USE*WHEN*WITH*1*
2*3*4*5*6*7*8*9*0*X*Y*COMPUTER,FISH,PLOT**

A stop word list can be devised by running **VOCAB** on a portion of the file, and identifying all the insignificant words which appear. These words can be punched as above and used in subsequent **VOCAB** runs.

ACKNOWLEDGEMENTS

The help and encouragement of Mr. Robert Russell of the Pacific Southwest Forest and Range Experiment Station in sending us the original programs is gratefully acknowledged. Later, he spent several hours, telephone in hand, helping us through the initial pangs and intricacies of FAMULUS. Mr. R. J. Olson and John R. Tadlock introduced the programming changes necessary to run the programs on our local CDC computer. They also implemented modifications which increased program capacity and flexibility necessary as our data file expanded in size. Bouquets go to Ms. Judy Helbling for her assistance and patience in editing this document.

REFERENCES

Burton, H. D. and T. B. Yerke, 1969, FAMULUS: A computer-based system for augmenting personal documentation efforts, pp. 53-61. Proceedings of the American Society for Information Science, Vol. 6.

Schultz, V., L. L. Eberhardt, J. M. Thomas, and M. L. Cochran, 1976, A Bibliography of Quantitative Ecology. Dowden, Hutchinson & Ross, Inc. 384 pages.

Thomas, J. M., L. L. Eberhardt, and M. L. Cochran, 1975, Bibliography of mathematics and statistics in ecology, pp. 12-13. In: Pacific Northwest Laboratory Annual Report for 1974. BNWL-1950, Pt. 2. Battelle-Northwest, Richland, Washington.

U.S. Department of Agriculture, 1969, FAMULUS: A Personal Documentation System-User's Manual. Pacific Southwest Forest and Range Experiment Station, Berkeley, California. 39 pages.

APPENDIX A

CONTROL CARDS FOR EACH MAIN PROGRAM

APPENDIX A. Control Cards for Each Main Program*

Control Cards Accepted by Programs

Unless the control card has "(optional)" listed in the right-hand column, **it** is a necessary control card:

EDIT

First Run:

/ID/
/FIELDS/
/DESCRIPTOR FIELD/
/ORIGINAL/
/CITATIONS/

Update Run:

/ID/	
/NEW ID/	(optional)
/FIELDS/	(optional)
/DESCRIPTOR FIELD/	(optional)
/PRINT/	(optional)
/DELETE/	(optional)
/REPLACE/	(optional)
/CITATIONS/	(optional)

SORT

/ID/	
/FIELDS/	
/SELECT/	(optional)

---

*U.S. Dept. Agriculture, 1969.

MERGE

/ID/	
/ID/	
/NEW ID/	(optional)
/PRINT/	(optional)

GALLEY

/ID/	
/WIDTH/	(optional)
/SELECT/	(optional)
/FIELD CHOICE/	(optional)
/PRINT BY FIELDS/	(optional)
/NO NUMBERS/	(optional)
/NUMBERS/	(optional)

INDEX

/ID/	
/SELECT/	(optional)
/PUNCH/	(optional)

VOCAB

/ID/	
/FIELDS/	(optional)
/SELECT/	(optional)
/VOCABULARY/	

SEARCH

/ ID/	
/FIELDS/	(optional)
/SELECT/	(optional)
/NUMBERS/	(optional)
/WIDTH/	(optional)
/FIELD CHOICE/	(optional)

OSSIFY

/ID/	
/SELECT/	(optional)

APPENDIX B

PROGRAMMER'S AIDS AND SPECIAL PROGRAMS

APPENDIX B. Programmer's Aids and Special Programs

This example shows how several programs within the FAMULUS System can be run as one job. Cards flagged with A through G in the left margin are computer control cards whereas the rest are program control cards. The main program call cards (EDIT, SORT, GALLEY, INDEX) are shown with their corresponding control cards in the order necessary to successfully execute this job. The use of this sequence would enable the user to enter his data, create a file, sort **it** into alphabetical order, catalog **it** as a permanent file, and have an index of the descriptor words in one continuous listing. As the user becomes more familiar with the FAMLIB System, he will see where this can be done with several combinations of programs. We hope this figure aids those who wish to combine several jobs into one run.

ERROR MESSAGES

EDIT

The two most frequent messages we encountered were:

/ID/ DOES NOT MATCH ID ON INPUT TAPE

and

FIELDS ARE OUT OF ORDER SKIPPING THIS CITATION

The first message needs no explanation, but the second one occurred for two reasons. Either input was being attempted in a manner inconsistent with the order defined on the FIELDS card or the blank card required between each data set was missing. The entire set of unentered cards will be set off with asterisks on the output.

Three other error conditions which occurred were:

1. ERROR IN ABOVE STATEMENT

This refers to a control card error immediately preceding the message.

2. THIS STATEMENT DOES NOT BELONG IN EDIT

An illegal program control card was used.

3. NEW FIELD LABELS DIFFER IN NUMBER FROM THE OLD ONES

In renaming the fields, either extra labels were included or some were omitted.

Other error messages are self-explanatory.

SORT

The most frequent message was:

/ID/ DOES NOT MATCH ID ON INPUT TAPE

The other messages result from erroneous use of the /FIELDS/ control card as follow:

- Control card missing
- Illegal fields
- Incorrectly punched card

MERGE

Once again, the most frequent message was:

/ID/ DOES NOT MATCH ID ON INPUT TAPE

The /ID/ statement on both files will be checked and the program will indicate which file /ID/ is incorrect.

The program will print an error message **if it** encounters any part of the tape which is not in the correct **SORT** order, and cease any further processing.

GALLEY

The /ID/ error message was again most frequent. Other error messages are self-explanatory.

INDEX

The major error message occurred because the /DESCRIPTOR FIELD/ was incorrectly defined.

VOCAB

DICTIONARY LIMITS EXCEEDED is the most frequent error message encountered when using this program. The program prints the number associated with the last set of reference data checked so the user has only to rerun this program using the /SELECT/ option, beginning at the number following the stop point.

SEARCH

The most frequent error occurs when key punching the search formulas used by this program. Incorrectly punched formulas or failure to balance the parentheses results in an error message with clear directives as to needed action. The error halts any further processing by this program.

OSSIFY

Errors in punching the numbers on the /SELECT/ control card were the most frequent.

CENTRAL MEMORY REQUIREMENTS FOR FAMULUS PROGRAMS ON CYBER 70*

EDIT	72K
GALLEY	60K
INDEX	146K
MERGE	65K
OSSIFY	55K
SEARCH	60K
SORT	100K
VOCAB	202K

---

*Control Data Cyber 70 Computer System, Boeing
Computer Services, Richland, WA

LOCATION OF TAPES ON CONTROL CARDS FOR PROGRAM CALLS

DATA INPUT ON TAPE 1

EDIT(INPUT,OUTPUT,TAPE1,TAPE2)   also   EDIT(,FILENAM,FILENAM)

GALLEY(INPUT,OUTPUT,TAPE1)   also   GALLEY(, ,FILENAM)

INDEX(INPUT,OUTPUT,PUNCH,TAPE1)   also   INDEX(,,FILENAM)

[illegible]

OSSIFY (INPUT,OUTPUT,PUNCH,TAPE1) also OSSIFY(,,FILENAM)

SEARCH(INPUT,OUTPUT,TAPE1,TAPE2) also SEARCH(,,FILENAM)

`SORT(INPUT,OUTPUT,TAPE1,TAPE2)` also `SORT(,FILENAM,FILENAM)`

VOCAB(INPUT,OUTPUT ,PUNCH,TAPE1)   also   VOCAB(,,TAPE1)

SYSTEM AND CONTROL CARDS USED TO PRODUCE AN 8 1/2 x 11 INCH OUTPUT

	USER'S NAME	LOCATION
GREEN,CM60000,T600,I0600.		
ACCOUNT(CUB438,CUB438)		
FTN(SYSEdit,R=3)		
PRNTLMT(55555)		
ATTACH(FILENAM,ID=FAMULUS)		
ATTACH(FAMLIB,ID=FAMULUS)		
LIBRARY(FAMLIB)		
MAP(OFF)		
LOAD(LGO)		
SEARCH(,FILENAM)		
==END-OF-RECORD-CARD==		

We produced a book which was called "GREEN" consisting of citations defined by categories in the **CHK** field. We used the control cards above subroutine PRETTY (page B.9) and the program SEARCH control cards on page B.14. The dimensions of the book pages were 8-1/2 by 11 inches.

The normal output of citations for program SEARCH is:

10. INNIS, G. 1974. DYNAMIC ANALYSIS IN "SOFT SCIENCE" STUDIES: IN DEFENSE OF DIFFERENCE EQUATIONS. PP. 102-122. IN, MATHEMATICAL PROBLEMS IN BIOLOGY. LECTURE NOTES IN MATHEMATICS 2. (P. VAN DEN DRIESSCHE, ED.). SPRINGER-VERLAG NEW YORK INC., NEW YORK, NEW YORK LOTKA-VOLTERRA SYSTEM MATH-MODEL

Subroutine PRETTY arranged citations in a preset order as follows:

1173. FREDRICKSON, A. G. 1971. A MATHEMATICAL THEORY OF AGE STRUCTURE IN SEXUAL POPULATIONS: RANDOM MATING AND MONOGAMOUS MARRIAGE MODELS. MATH. BIOSCI. 10, 117-143, AGE ZERO1

This allowed us to have the citation number set apart in the margin of the book.

Subroutine PSPACE caused the last page of "GREEN" to be numbered near the bottom of the page, otherwise the page number was on the fourth line down from the last line of the last citation in that category.

Subroutine PAGER allows the user to preset the first page number. **It** counts the number of lines per citation prior to printing and checks that number against the number of lines remaining on the page, thus preventing the citations from overlapping onto the next page. On the last page when an incomplete page is printed, the page number will print normally, at the bottom of the page.

A page from "Green" is shown as Fig. B1.

```

      SUBROUTINE PRETTY(CIT,LCIT,INDENT,LEN,FCHOI,JLAB,ICHOI)      07/18/75
C PRINTS A CITATION IN BIBLIOGRAPHIC FORMAT,                  PRETTY 3
C THE CITATION IS AT CIT,                                     PRETTY 4
C THE CITATION NUMBER IS LCIT, WHICH IS PRINTED ON THE LEFT UNLESS IT IS PRETTY 5
C 0 IN WHICH CASE IT IS NOT PRINTED,                          PRETTY 6
C INDENT=1 SAYS TO INDENT ALL LINES BUT THE FIRST,           PRETTY 7
C INDENT=0 SAYS TO INDENT NONE OF THEM,                       PRETTY 8
C THE FIELDS ARE PRINTED IN THE ORDER DICTATED BY THE IFIELD ARRAY, PRETTY 9
C LEN IS THE PAGE WIDTH, IN CHARACTERS, IT IS THE MAX. NUMBER OF PRETTY10
C CITATION TEXT CHARACTERS ON A LINE PLUS 4 CHARACTERS FOR LCIT(IF LCIT PRETTY11
C IS NOT 0),                                                  PRETTY12
      COMMON /CORE/NCHW,BLANKS,BLANK                          PRETTY13
      INTEGER BLANKS,BLANK                                    PRETTY14
      COMMON /FIEL/FIELD(10),IFIELD(10),NFIELD,KDESCR        PRETTY15
      INTEGER FIELD                                           PRETTY16
      INTEGER CIT(3,2),LINE(16)                              PRETTY17
C THE DIMENSION OF LINE() IS LENLIN+1.                      PRETTY18
      LOGICAL GOLINE,GOCHAR,ENTROP,BLOCK,DONE                PRETTY19
      LOGICAL FCHOI                                          07/18/76
      INTEGER JLAB(10)                                       07/18/77
      DATA LENLIN/15/                                       PRETTY20
C INITIALIZE                                                  PRETTY21
      GOLINE=.FALSE.,                                         PRETTY22
      KF=0                                                    PRETTY23
      NC=0                                                    PRETTY24
      LL=0                                                    PRETTY25
      ENTROP=LCIT,EQ,0                                         PRETTY26
      BLOCK=INDENT,LE,0                                        PRETTY27
      MOST=MIN0(LEN-7,LENLIN*NCHW)                            PRETTY28
      IF(ENTROP) MOST=LEN                                       PRETTY29
      DONE=.FALSE.,                                           PRETTY30

C
C-----PAGER(-NO,) IS THE DESIGNATION FOR NUMBER OF LINES BETWEEN CITATIONS
C      THIS HAS TO BE A MINUS NUMBER
C
C-----NEED MORE LINES BETWEEN CITATIONS FOR VINCE SCHULTZ TO DO READINGS
C
      CALL PAGER(-9) *
C PREPARE TO START A NEW LINE,                                PRETTY32
      10 CALL MOVE(BLANKS,LINE,-LENLIN)                       PRETTY33
      GOCHAR=.FALSE.,                                         PRETTY34
      NL=MOST-LL                                              PRETTY35
C TEST IF CURRENT FIELD DEPLETED AND IF SO, GET NEXT FIELD, PRETTY36
      20 IF(NC,GT,0) GO TO 30                                  PRETTY37
      IF(FCHOI) GO TO 25                                       07/18/78

```

```

      IF(KF,GE,NFIELD) GO TO 80
      KF=KF+1
      KFF=IEIELD(KFI)
      GO TO 27
25 IF(KF,GE,NCHW ) GO TO 80
      KF=KF+1
      KFF=JLAB(KF)
27 NC=CIT(1,KFF)
      LC=CIT(2,KFF)
      IF(.NOT.(GOCHR,AND,NC,GT,0)) GO TO 20
      N=MING(NL,3)
      CALL CHOP(BLANKS,0,LINE,LL,-N)
      LL=LL+N
      NL=NL-N
      GO TO 20
C SELECT GROUP OF CHARS TO MOVE INTO LINE WAITING TO PRINT.
30 CALL SCOUT(0,0,BLANK,CIT,LC,NC,M)
      LC=LC+M
      NC=NC-M
      N=MIND(NC,NL+1)
      IF(NC,LE,NL) GO TO 35
      CALL SCOUT(1,1,BLANK,CIT,LC,N,M)
      IF(M,EQ,N,AND,NOT,GOCHAR) M=1
      N=N-M
35 IF(N,LE,0) GO TO 40
      CALL CHOP(CIT LC,LINE,LL,N)
      LC=LC+N
      NC=NC-N
      LL=LL+N
      NL=NL-N
      GOCHAR=.TRUE.
      GO TO 20
C PRINT THE LINE WAITING.
40 CALL PAGER(1)
      IF(ENTROP) GO TO 60
      IF(GOLINE) GO TO 50
C---PRINT FIRST LINE WITH CITATION NUMBER
      GOLINE=.TRUE.
      PRINT 1000,LC T,(LINE(J),J=1,LENLIN

```

```

PRETTY38
PRETTY39
PRETTY40
07/18/79
07/18/10
07/18/11
07/18/12
07/18/13
PRETTY42
PRETTY43
PRETTY44
PRETTY45
PRETTY46
PRETTY47
PRETTY48
PRETTY49
PRETTY50
PRETTY51
PRETTY52
PRETTY53
PRETTY54
PRETTY55
PRETTY56
PRETTY57
PRETTY58
PRETTY59
PRETTY60
PRETTY61
PRETTY62
PRETTY63
PRETTY64
PRETTY65
PRETTY66
PRETTY67
PRETTY68
PRETTY69
07/18/14
PRETTY70
PRETTY71

```

```

C
C-----SPECIAL FORMAT WITH NUMBERS FOR GREEN BOOK - 12/3/75/M COCHRAN
C
1000 FORMAT (2X,I4,"","1X,15A8)
C
      GO TO 70
C---PRINT REMAINING LINES UNDER COMMON FORMAT
      50 PRINT 2000, (LINE(J),J=1,LENLIN)
      2000 FORMAT (12X,15A8)
      GO TO 70
C---PRINT LINE WITHOUT CITATION NUMBER WITH FIRST LINE OFFSET  RJ OLSON
      60 PRINT 3000, (LINE(J),J=1,LENLIN)
      ENTROP = ,FALSE,
      GOLINE = ,TRUE,

      3000 FORMAT( 8X,15A8)
C RESET FOR NEXT LINE, IF ANY,
      70 IF(DONE) GO TO 100
      LL=5
      IF(BLOCK) LL=0
      GO TO 10
C FINISHED WITH CITATION, BUT MUST PRINT PARTIAL LAST LINE.
      80 DONE=,TRUE,
      GO TO 40
C ALL DONE, RETURN,
      100 RETURN
      END
      SUBROUTINE PSPACE (NN)

C
C      SPECIAL ROUTINE FOR BOOK PRINTING ( GREEN)
C
C      IF NN IS GREATER THAN ZERO, THEN SPACE THE PRINTER (NN) LINE
C      IF NN IS NOT GREATER THAN ZERO RETURN
C
10      IF ( NN ,LE, 1 ) GO TO 20
      PRINT 2000
2000  FORMAT ( 1H0 )
      NN = NN - 2
      GO TO 10
20      IF ( NN ,EQ, 1 ) PRINT 1000
1000  FORMAT (1X)
      RETURN
      END
      SUBROUTINE PAGER (N)

```

```

PRETTY73
07/18/17
PRETTY74
07/18/18
PRETTY76
07/18/19
PRETTY77
07/18/20
07/18/21
07/18/22
PRETTY79
PRETTY80
PRETTY81
PRETTY82
PRETTY83
PRETTY84
PRETTY85
PRETTY86
PRETTY87
PRETTY88
PRETTY89

```


SUBROUTINE PAGER (N)

COMMON/PAGING/IPAGE,LINES,MAXLIN,PAGHED(13)

DATA IPAGE,LINES,MAXLIN,(PAGHED(J),J=1,13)/0,0,58,13*1H /

START PAGE NUMBER AT -2 BECAUSE OF HEADER PAGES AND
PUT PAGE NUMBER ON LAST HEADER PAGE

DATA JPAGE / -2 /

TO SET UP PAGE NUMBERS, SUBTRACT FROM HEADING PAGES, PUT THAT NUMBER
AFTER THE PLUS OR MINUS WHICHEVER IS THE CASE

DATA LONGCIT/7/

ALLOWS FOR LENGTH OF CITATION TO 7 LINES; OTHERWISE IT MAY
OVERLAP ONTO THE NEXT PAGE

IF (N, EQ, 0) GO TO 50

IF (N, GT, 0) GO TO 30

NN = -N

LINES = LINES - NN

IF(LINES,LT,LONGCIT) GO TO 50

C-C-C-IF (LINES, LE, 0) GO TO 50

SPACE PRINTER - NN LINES BETWEEN CITATIONS, ETC., .

CALL PSPACE (NN)

RETURN

PAGER CALLED WITH POSITIVE NUMBER; ADJUST LINE COUNT

30 LINES = LINES + N

IF NEW PAGE NOT NEEDED - RETURN

IF (LINES, GT, 0) RETURN

LINES = MAX0 (0, MAXLIN - N)

JPAGE = JPAGE + 1

6000 FORMAT(1H0/T30,15,"","/1"/)

C--- (1H0/T30,15, CAUSES 3 BLANK BEFORE PAGE-NUMBER

/1","0","0") CAUSES PRINTING TO START DOWN 5 LINES

ADD BLANK LINES AT BOTTOM OF PAGE TO LINE UP PROPERLY

C

```

CALL PSPACE ( JLINES )
PRINT 6000 , JPAGE
RETURN

```

C

```

50 JLINES = LINES + 1
   LINES = 0
100 RETURN
END

```

```

X00 789 END--OF--RECORD--CARD 7/8/9 *****

```

```

/ID/BIBLIOGRAPHY

```

```

/SELECT/(1-4700)

```

```

/WIDTH/(64)

```

```

/FIELD CHOICE/(AUTH, YEAR, TITL, REF, CHK, DESC)

```

```

/FIELDS/(CHK)

```

```

/SEARCH/(AGE)

```

```

/SEARCH/(CAPT-RECAPT)

```

```

/SEARCH/(COMP-MODEL)

```

```

/SEARCH/(COMPET-DEN)

```

```

/SEARCH/(COMPUTERS)

```

```

/SEARCH/(CYCLES)

```

```

/SEARCH/(DIVERSITY)

```

```

/SEARCH/(ENERGET)

```

```

/SEARCH/(EXPL-MNG)

```

```

/SEARCH/(FREQ-DIST)

```

```

X00 ==END OF RECORD CARD=====

```

```

W      0/6/7/8/9      MOHAWK UT-200 END-OF-JOB CARD

```

```

W      0/6/7/8/9      MOHAWK UT-200 END-OF-JOB CARD

```

51. ALLEN, K. R. 1966. DETERMINATION OF AGE DISTRIBUTION FROM AGE - LENGTH KEYS AND LENGTH DISTRIBUTIONS, IBM 7090, 7094 FORTRAN IV. TRANS. AMER. FISH. SOC. 95(2), 230-231, AGE
63. ALLEN, R. L. AND P. BASASIBWAKI, 1974. PROPERTIES OF AGE STRUCTURE MODELS FOR FISH POPULATIONS. J. FISH. RES. BOARD CAN. 31(6), 1119-1125. AGE SIMULATION; STOCK-RECRUITMENT; STABILITY; RICKER MODEL
73. ANDERSEN, F. S. 1960. COMPETITION IN POPULATIONS CONSISTING OF ONE AGE GROUP. BIOMETRICS 16(1), 19-27, AGE
239. BARTLETT, M. S. 1970. AGE DISTRIBUTIONS. BIOMETRICS 26(3), 377-385. AGE
561. CAUGHLEY, G. 1974. INTERPRETATION OF AGE RATIOS. J. WILDL. MANAGE. 38(3), 557-562. AGE COMPUTER; SIMULATION; MODEL
563. CAUGHLEY, G. AND J. CAUGHLEY. 1974. ESTIMATING MEDIAN DATE OF BIRTH. J. WILDL. MANAGE. 38(3), 552-556. AGE PROBIT ANALYSIS
733. COUTINHO, A. B. AND F. A. COUTINHO. 1968. THE CONSERVATION OF THE BIOMASS DENSITY AND THE AGE STRUCTURE OF THE POPULATIONS. BULL. MATH. BIOPHYS. 30, 553-563, AGE
829. DAPSON, R. W. 1971. QUANTITATIVE COMPARISON OF POPULATIONS WITH DIFFERENT AGE STRUCTURES. ANN. ZOOL. FENN. 8(1), 75-79. AGE PROBABILITY; CONFIDENCE INTERVAL
939. DOI, T. 1949. AN INVESTIGATION ON THE AGE COMPOSITION OF FISH STOCK BY THE ANALYSIS OF REGRESSION. BULL. JAPANESE SOC. SCI. FISH. 15(7), 306-310. AGE SIMPLE LINEAR REGRESSION
990. EBERHARDT, L. AND R. I. BLOUCH. 1955. ANALYSIS OF PHEASANT AGE RATIOS. TRANS. N. AMER. WILDL. CONF. 20, 357-367, AGE
1035. EMLEN, J. M. 1970. AGE SPECIFICITY AND ECOLOGICAL THEORY. ECOLOGY 51(4), 588-601, AGE
1173. FREDRICKSON, A. G. 1971. A MATHEMATICAL THEORY OF AGE STRUCTURE IN SEXUAL POPULATIONS; RANDOM MATING AND MONOGAMOUS MARRIAGE MODELS. MATH. BIOSCI. 10, 117-143, AGE ZERO1

FIGURE B1. A Page from "Green" as 8 1/2 x 11 inch Output

APPENDIX C
PROGRAM LISTINGS

4

APPENDIX C
PROGRAM LISTINGS

APPENDIX C. Program Listings

<pre> PROGRAM EDIT(INPUT,OUTPUT,TAPE1,TAPE2,TAPE3, , TAPE5(=INPUT,TAPE51=OUTPUT) C EDIT PROGRAM, PART OF THE FAMULUS SYSTEM. COMMON /CORE/NCHW,BLANKS,BLANK INTEGER BLANKS,BLANK COMMON /F1IN/INTAPE COMMON /FAM/SLASH,STAR,COMMA,ZERO,MXCHAR,MXLCES,MXWDES,1XCHW, , MXFREE,MXWFERE INTEGER SLASH,STAR,COMMA,ZERO COMMON /FIEL/FIEL(10),IFIEL(10),NFIELD,KDESCR INTEGER FIEL COMMON /IPT/CARD(10),KOLUM,HOLD,ENDCAR,BLACAR,CONCAR,ERRIN,NWC, , MXFL,MXCAR,NCARIN,KHR,KIN,LIT,LITX,SYM(5),NUM,NCSYM,CATLOG, , KARA,KARB INTEGER CARL,SYM LOGICAL HOLD,ENDCAR,BLACAR,CONCAR,ERRIN,CATLOG COMMON /PAGING/IPAGE,LINES,MAXLIN,PAGHED(13) INTEGER PAGHED COMMON /TEST/ITEST COMMON /HIT/NREP,LREP(2,100),KREP(100),PATREP(800),LOCPAT(100), , NCHPAT(100),LOCREP(100),NCHREP(100) INTEGER PATREP INTEGER ID(13),IDX(13),IDY(13),FLDX(10),IFLDX(10),LDEL(500) INTEGER LPRINT(100),MAYBE(64),DESCR,CRD(10,65) INTEGER CITX(3,180),CTX(540),CIT(3,180),CT(540) EQUIVALENCE (CITX,CTX),(CIT,CT) INTEGER *CHPT(100),NCHRP(100),LOCPT(100),LOCRP(100) LOGICAL MATCH,MEMBER,ORIG,ENDZ,ENLX,PRIN,TRK,SOOF,CATLG KOT(N)=(N+NCHW-1)/NCHW CALL USHER(ID,KERR) Kt=1 IF(KERR.NE.0) GO TO 900 MTX=1 MTY=2 LEN=127 NFIELD=0 NIEL=0 LDEL(1)=0 MXDEL=500 NREP=0 LPRINT=0 ORIG=.FALSE. ENDZ=.TRUE. ENDX=.TRUE. CATLG=.FALSE. DESCR=BLANKS KDESCR=0 LCITX=0 LCITZ=0 LCITY=0 CALL CHOP(ID,0,IDY,0,MXFREE) C STATEMENT INPUT, 10 KE=3 IF(ERRIN) GO TO 900 IF(ITEST.NE.0) CALL PAGER(-4) CALL SETSYM IF(ENDCAR) GO TO 150 </pre>	<pre> EDIT 2 EDIT 3 EDIT 4 EDIT 5 EDIT 6 EDIT 7 EDIT 8 EDIT 9 EDIT 10 EDIT 11 EDIT 12 EDIT 13 EDIT 14 EDIT 15 EDIT 16 EDIT 17 EDIT 18 EDIT 19 EDIT 20 EDIT 21 EDIT 22 EDIT 23 EDIT 24 EDIT 25 EDIT 26 EDIT 27 EDIT 28 EDIT 29 EDIT 30 EDIT 31 EDIT 32 EDIT 33 EDIT 34 EDIT 35 EDIT 36 EDIT 37 EDIT 38 EDIT 39 EDIT 40 EDIT 41 EDIT 42 EDIT 43 EDIT 44 EDIT 45 EDIT 46 EDIT 47 EDIT 48 EDIT 49 EDIT 50 EDIT 51 EDIT 52 EDIT 53 EDIT 54 EDIT 55 EDIT 56 EDIT 57 EDIT 58 </pre>

CALL FABRIC(KON)	EDIT	59
KE=2,	EDIT	60
IF (ERRIN) GO TO 900	EDIT	61
IF (KOLUM.GT.1) CALL DVEST	EDIT	62
C /WIDTH/	EDIT	63
IF (KON.NE.15) GO TO 20	EDIT	64
CALL GETNUM(M,1,1,N,KERR)	EDIT	65
KE=21	EDIT	66
IF (KERR.NE.0) GO TO 900	EDIT	67
LEN=MAX0(MIN0(M,127),20)	EDIT	68
GO TO 10	EDIT	69
20 CONTINUE	EDIT	70
C /ORIGINAL/	EDIT	71
IF(KON.NE.9) GO TO 30	EDIT	72
ORIG=.TRUE,	EDIT	73
CALL SPOT(ORIG,0,0,0,4HLIII,24HORIG	EDIT	74
GO TO 10	EDIT	75
C /NEW ID/	EDIT	76
30 IF(KON.NE.4) GO TO 40	EDIT	77
CALL GOSSIP(IDY)	EDIT	78
KE=18	EDIT	79
IF(ERRIN) GO TO 900	EDIT	80
CALL CHOP(IDY,0,PAGHD,0,MXFREE)	EDIT	81
GO TO 10	EDIT	82
C /FIELDS/	EDIT	83
40 IF(KON.NE.2) GO TO 50	EDIT	84
CALL GETLAB(FIELD,1,10,NFIELD,KERR)	EDIT	85
CALL SPOT(NFIELD,KERR,0,0,4HIIII,24H IFIELDKERR	EDIT	86
KE=4	EDIT	87
IF(KERR.EQ.1) GO TO 900	EDIT	88
KE=5	EDIT	89
IF(KERR.EQ.2) GO TO 900	EDIT	90
CALL TALUD(SHFIELD,FIELD,0,NCHW*NFIELD)	EDIT	91
DO 45 K=1,NFIELD	EDIT	92
45 IFIELD(K)=K	EDIT	93
GO TO 10	EDIT	94
C /PRINT/	EDIT	95
50 IF(KON.NE.5) GO TO 60	EDIT	96
CALL GETNUM(LPRINT,1,100,NPRINT,KERR)	EDIT	97
KE=6	EDIT	98
IF(KERR.EQ.1) GO TO 900	EDIT	99
KE=7	EDIT	100
IF(KERR.EQ.2) GO TO 900	EDIT	101
CALL KORAN(6HLPRINT,LPRINT,NPRINT+1)	EDIT	102
"3 TO 10	EDIT	103
C /DELETE/	EDIT	104
60 IF(KON.NE.7) GO TO 70	EDIT	105
CALL GETNUM(LDEL,1,MXDEL,NDEL,KERR)	EDIT	106
KE=8	EDIT	107
IF(KERR.EQ.1) GO TO 900	EDIT	108
KE=9	EDIT	109
IF(KERR.EQ.2) GO TO 900	EDIT	110
CALL KORAN(4HLDL,LDEL,NDEL+1)	EDIT	111
GO TO 10	EDIT	112
C /DESCRIPTOR FIELD/	EDIT	113
70 IF(KON.NE.13) GO TO 80	EDIT	114
	EDIT	115

CALL GETLAB(DESCR,1,1,Q,KERR)	EDIT 116
KE=10	EDIT 117
IF(KERR.EQ.1) GO TO 900	EDIT 118
KE=11	EDIT 119
IF(KERR.EQ.2) GO TO 900	EDIT 120
CALL TALNUD(5HDESCR,DESCR,0,NCHW)	EDIT 121
GO TO 10	EDIT 122
C /REPLACE/	EDIT 123
80 IF(KON.NE.8) GO TO 90	EDIT 124
CALL TINKER	EDIT 125
KE=19	EDIT 126
IF(ERRIN) GO TO 900	EDIT 127
GO TO 13	EDIT 128
C /CATALOG/	EDIT 129
C THE /CATALOG/ STATEMENT IS ONLY FOR COMPATIBILITY WITH THE CITATION	EDIT 130
C CARD FORMAT OF THE OBSOLETE CATALOG SYSTEM,	EDIT 131
99 IF(KON.NE.16) GO TO 100	EDIT 132
CATLG=.TRUE.	EDIT 133
CALL KORAN(5HCATLG,0,0)	EDIT 134
GO TO 10	EDIT 135
C /CITATIONS/	EDIT 134
100 KE=12	EDIT 137
IF(KON.NE.6) GO TO 900	EDIT 138
ENDZ=.FALSE.	EDIT 139
CATLOG=CATLG	EDIT 140
IF(CATLOG) BLACAR=.TRUE.	EDIT 141
IF(.NOT.ENDCAR) GO TO 110	EDIT 142
INTAPE=3	EDIT 143
CALL REWAND(INTAPE)	EDIT 144
ENDCAR=.FALSE.	EDIT 145
CALL INCARD	EDIT 146
110 CALL SPOT(CATLOG,BLACAR,ENDCAR,INTAPE,4HLLLI,	EDIT 147
, 24HCATLOGBLACARENDCARINTAPE)	EDIT 148
C FINALIZE STATEMENT INPUT,	EDIT 149
150 IF(.NOT.ORIG) GO TO 180	EDIT 150
C ORIGINAL HUN,	EDIT 151
ENDX=.TRUE.	EDIT 152
KE=13	EDIT 153
IF(NFIELD.LE.0.OR.ENDZ) GO TO 900	EDIT 154
KE=14	EDIT 155
IF(NDEL+NPRINT+NREP.NE.0) GO TO 903	EDIT 156
DO 160 K=1,NFIELD	EDIT 157
160 IFIELD(K)=K	EDIT 158
GO TO 200	EDIT 159
C NOT ORIGINAL RUN,	EDIT 160
180 ENDX=.FALSE.	EDIT 161
CALL IOID(0,MTX,IDX,FLDX,IFLDX,NFLDX,KDES)	EDIT 162
CALL PAGER(-5)	EDIT 163
CALL IDPRIN(IDX,FLDX,IFLDX,NFLDX,KDES,10,10HINPUT TAPE)	EDIT 164
CALL SIMILE(1,ID,0,MXFREE,IDX,0,MXFREE,M)	EDIT 165
KE=16	EDIT 166
IF(M.EQ.0) GO TO 900	EDIT 167
KE=17	EDIT 168
IF(NFIELD.GT.0.AND.NFIELD.NE.NFLDX) GO TO 900	EDIT 169
IF(NFIELD.GT.0) GO TO 185	EDIT 170
NFIELD=NFLDX	EDIT 171
CALL MOVE(FLDX,FIELD,NFIELD)	EDIT 172

185	CALL MOVE(IFLDX,IFIELD,NFIELD)	EDIT 173
	IF(NREP.LE.0) GO TO 192	EDIT 174
	DO 190 J=1,NREP	EDIT 175
	K=LOOK(FIELD,NFIELD,KREP(J))	EDIT 176
	KF=20	EDIT 177
	IF(K.GT.NFIELD) GO TO 900	EDIT 178
190	KREP(J)=K	EDIT 179
192	IF(MATCH(DESCR,BLANKS),AND,KDES.GT.0) DESCR=FIELD(KDES)	EDIT 180
C	INITIALIZE CITATION PROCESSING.	EDIT 181
200	CONTINUE	EDIT 182
	KDESCR=0	EDIT 183
	L=LOOK(FIELD,NFIELD,DESCR)	EDIT 184
	KF=15	EDIT 185
	IF(L.GT.NFIELD,AND,.NOT.MATCH(DESCR,BLANKS)) GO TO 900	EDIT 186
	IF(L.LE.NFIELD) KDESCR=L	EDIT 187
	LCITY=0	EDIT 188
	CALL PAGER(-2)	EDIT 189
	CALL IDPRINC(IDY,FIELD,IFIELD,NFIELD,KDESCR,11,11HOUTPUT TAPE)	EDIT 190
	CALL IOID(1,MTY,IDY,FIELD,IFIELD,NFIELD,KDESCR)	EDIT 191
C	PROCESS CITATIONS FROM INPUT TAPE,	EDIT 192
	IF(ENDX) GO TO 310	EDIT 193
210	LCITX=LCITX+1	EDIT 194
	IF(ITEST.NE.0) CALL PAGER(-2)	EDIT 195
	CALL KORAN(5HLCITX,LCITX,1)	EDIT 196
	PRIN=.FALSE.	EDIT 197
	IOX=0	EDIT 198
	IF(MEMBER(LDEL,NDEL,LCITX)) IOX=-1	EDIT 199
	IF(NREP.LE.0) GO TO 225	EDIT 200
	TINK=.TRUE.	EDIT 201
	DO 220 J=1,NREP	EDIT 202
	IF(LCITX.GE.LREP(1,J),AND,LCITX.LE.LREP(2,J)) GO TO 230	EDIT 203
220	CONTINUE	EDIT 204
225	TINK=.FALSE.	EDIT 205
230	CONTINUE	EDIT 206
	IF(TINK) CALL IOTAPE(IOX,MTX,CIT,LASTX)	EDIT 207
	IF(.NOT.TINK) CALL IOTAPE(IOX,MTX,CITX,LASTX)	EDIT 208
	ENDX=LASTX,EQ,1	EDIT 209
	IF(ENDX) GO TO 300	EDIT 210
	IF(IOX.EQ.-1) GO TO 210	EDIT 211
	IF(MEMBER(LPRINT,LPRINT,LCITX)) PRIN=.TRUE.	EDIT 212
	IF(.NOT.TINK) GO TO 290	EDIT 213
C	DO REPLACEMENTS.	EDIT 214
	LCY=CIT(2,1)	EDIT 215
	DO 280 KF=1,NFIELD	EDIT 216
	IF(ITEST.NE.0) CALL PAGER(-2)	EDIT 217
	NC=CIT(1,KF)	EDIT 218
	LC=CIT(2,KF)	EDIT 219
	NART=CIT(3,KF)	EDIT 220
	CALL SPOT(KF,NC,LC,NART,4HIIII,24HKF NC LC YAHT)	EDIT 221
	JJ=0	EDIT 222
	DO 240 J=1,NREP	EDIT 223
	IF(.NOT.(LCITX.GE.LREP(1,J),AND,LCITX.LE.LREP(2,J),AND,KF.EQ.,	EDIT 224
	KREP(J))) GO TO 240	EDIT 225
	JJ=JJ+1	EDIT 226
	NCHPT(JJ)=NCHPAT(J)	EDIT 227
	NCHRP(JJ)=NCHREP(J)	EDIT 228
	LOCPT(JJ)=LOCPAT(J)	EDIT 229

LOC RP(JJ)=LOC REP(J)	EDIT 230
240 CONTINUE	EDIT 231
NRP=JJ	EDIT 232
CALL KORAN(5HNCHPT,NCHPT,NRP)	EDIT 233
CALL KORAN(5HNCHRP,NCHRP,NRP)	EDIT 234
CALL KORAN(5HLOCPT,LOCPT,NRP)	EDIT 235
CALL KORAN(5HLOC RP,LOC RP,NRP)	EDIT 236
IF(NRP.GT.0) GO TO 250	EDIT 237
C NO REPLACEMENTS IN THIS FIELD.	EDIT 238
LLX=KUT(LC)+1	EDIT 239
LLY=KUT(LCY)+1	EDIT 240
CALL MOVE(CT(LLX),CTX(LLY),KUT(NC))	EDIT 241
CALL TALMUD(FIELD(KF),CTX(LLY),0,NC)	EDIT 242
NCY=NC	EDIT 243
GO TO 270	EDIT 244
C REPLACEMENTS IN THIS FIELD.	EDIT 245
250 CALL MOVE(0,MAYBE,-64)	EDIT 246
DO 260 J=1,NRP	EDIT 247
IF(NCHPT(J).LE.0) GO TO 260	EDIT 248
KH=KHAR(PATREP,LOCPT(J)+1)+1	EDIT 249
MAYBE(KH)=1	EDIT 250
260 CONTINUE	EDIT 251
CALL ADAN(CIT,LC,NC,CITX,LCY,NCY,PATREP,LOCPT,LOC RP,NCHPT,NCHRP,	EDIT 252
NRP,MAYBE,PRIN,NART)	EDIT 253
270 CITX(1,KF)=NCY	EDIT 254
CITX(2,KF)=LCY	EDIT 255
CITX(3,KF)=NART	EDIT 256
LCY=LCY+KUT(NCY)*CHW	EDIT 257
CALL SPOT(NCY,LCY,NART,NRP,4HIIII,24HN CY LCY JART NRP)	EDIT 258
280 CONTINUE	EDIT 259
C CITATION IS NOW IN CITX AND READY TO GO,	EDIT 260
290 LCITY=LCITY+1	EDIT 261
IF(PRIN) CALL PRNCIT(CITX,LCITY,LCITX,LEN)	EDIT 262
CALL IOTAPE(1,MTY,CITX,0)	EDIT 265
GO TO 210	EDIT 264
C END OF INPUT TAPE,	EDIT 265
300 CALL REWAND(MTY)	EDIT 266
C PROCESS CITATIONS FROM CARDS,	EDIT 267
310 IF(ENDZ) GO TO 330	EDIT 268
C READ NEXT CITATION FROM CARDS,	EDIT 269
320 LCITZ=LCITZ+1	EDIT 270
CALL CARCIT(CIT,CRD,NCRD,GOOF)	EDIT 271
IF(NCRD.EQ.0) ENDZ=.TRUE,	EDIT 272
IF(ENDZ) GO TO 333	EDIT 273
IF(GOOF) GO TO 320	EDIT 274
LCITY=LCITY+1	EDIT 275
CALL PRNCIT(CIT,LCITY,0,LEN)	EDIT 276
CALL IOTAPE(1,MTY,CIT,0)	EDIT 277
GO TO 320	EDIT 278
C END OF INPUT CARD CITATIONS,	EDIT 273
330 CALL IOTAPE(1,MTY,CIT,1)	EDIT 280
CALL REWAND(MTY)	EDIT 281
C PRINT CONCLUDING MESSAGES,	EDIT 282
CALL PAGER(-51)	EDIT 263
CALL PAGER(1)	EDIT 284
PRINT 1000, LCITY	EDIT 285
1000 FORMAT(10H THERE ARE,15,30H CITATIONS ON THE OUTPUT TAPE,)	EDIT 286

CALL PAGER(-1)	EDIT 287
CALL PAGER(1)	EDIT 288
PRINT 2000	EDIT 289
2000 FORMAT(24H PROCESSING IS COMPLETE,)	EDIT 290
CALL PAGER(11)	EDIT 291
PRINT 8999	EDIT 292
8999 FORMAT(/	EDIT 293
1 /3(20X 17H XXYYX X X)	EDIT 294
2 /3(20X 17HX X X X)	EDIT 295
3 /3(20X 17HX X X X)	EDIT 296
4 /3(20X 17HX X X)	EDIT 297
5 /3(20X 17HX X X X)	EDIT 298
6 /3(20X 17HX X X X)	EDIT 299
7 /3(20X 17H XXXXX X X)	EDIT 300
8 //)	EDIT 301
STOP	EDIT 302
C ERROR MESSAGES.	EDIT 303
900 IF(.NOT.CONCAR) CALL 3IVEST	EDIT 304
CALL PAGER(2)	EDIT 305
IF(KE.EQ,21) GO TO 902	EDIT 306
GO TO (901,901,901,902,902, 902,902,902,902,902,	EDIT 307
902,903,904,904,905, 906,907,901,901,908),KE	EDIT 308
901 PRINT 9001	EDIT 309
9001 FORMAT(17H INPUT CARD ERROR,111(1H*))	EDIT 310
GO TO 950	EDIT 311
902 PRINT 9002	EDIT 312
9002 FORMAT(25H ERROR IN ABOVE STATEMENT,103(1H*))	EDIT 313
GO TO 950	EDIT 314
903 PRINT 9003	EDIT 315
9003 FORMAT(39H THIS STATEMENT DOES NOT BELONG IN EDIT,89(1H*))	EDIT 316
GO TO Y5P	EDIT 317
904 PRINT 9004	EDIT 318
9004 FORMAT(38H INPUT IS INCOMPATIBLE WITH /ORIGINAL/,90(1H*))	EDIT 319
GO TO 950	EDIT 320
905 PRINT 9005	EDIT 321
9005 FORMAT(54H DESCRIPTOR FIELD LABEL DOES NOT MATCH ANY FIELD LABEL,	EDIT 322
,74(1H*))	EDIT 323
GO TO 950	EDIT 324
906 PRINT 9006	EDIT 325
9006 FORMAT(37H /ID/ DOES NOT MATCH I1 ON INPUT TAPE,91(1H*))	EDIT 326
GO TO 951	EDIT 327
907 PRINT 9007	EDIT 328
9007 FORMAT(56H THE NEW FIELD LABELS DIFFER IN NUMBER FROM THE OLD ONES	EDIT 329
,72(1H*))	EDIT 330
GO TO 950	EDIT 331
908 PRINT 9008, KREP(J)	EDIT 332
9008 FORMAT(17H THE FIELD LABEL ,A4,53H IN /REPLACE/ DOES NOT MATCH ANY	EDIT 333
, OF YOUR FIELD LABELS,5X,49(1H*))	EDIT 334
950 PRINT 9050,KE	EDIT 335
9050 FORMAT(118X,6HEDIT 14)	EDIT 336
STOP	EDIT 337
C CIT(3,150) TEMPORARY STORAGE FOR A CITATION,	EDIT 338
C CITX(3,150) CURRENT CITATION FROM INPUT TAPE,	EDIT 339
C CRD(10,65) TEMPORARY STORAGE FOR CITATION IN CARD-IMAGE FORM,	EDIT 340
C CT(450) SAME AS CIT,	EDIT 341
C CTX(450) SAME AS CITX,	EDIT 342
C DESCR FIELD LABEL FROM /DESCRIPTOR FIELD/.	EDIT 343

C	DIFF	TRUE IFF A REPLACEMENT IS MADE.	EDIT 344
C	ENDX	LAST CITATION FROM INPUT TAPE HAS BEEN PROCESSED.	EDIT 345
C	ENDZ	LAST CITATION FROM CARDS HAS BEEN PROCESSED.	EDIT 346
C	FLDX(10)	FIELD LABELS READ FROM INPUT TAPE.	EDIT 347
C	GOOF	TRUE IFF INPUT CARD CITATION HAS ERROR.	EDIT 348
C	ID(15)	TEXT FROM /ID/.	EDIT 349
C	IDX(15)	TEXT FROM INPUT TAPE ID.	EDIT 350
C	IDY(15)	TEXT FROM /NEW ID/.	EDIT 351
C	IFLDX(10)	FIELD ORDER INDICATORS READ FROM INPUT TAPE.	EDIT 352
C	KDES	DESCRIPTOR FIELD READ FROM INPUT TAPE.	EDIT 353
C	KE	ERROR NUMBER.	EDIT 354
C	KREP(100)	THE J-TH REPLACEMENT APPLIES TO FIELD NUMBER KREP(J).	EDIT 355
C	LCITX	CURRENT CITATION NUMBER, INPUT TAPE.	EDIT 356
C	LCITY	CURRENT CITATION NUMBER, OUTPUT TAPE.	EDIT 357
C	LCITZ	CURRENT CITATION NUMBER, INPUT CARDS.	EDIT 358
C	LDEL(500)	DELETION LIST IN /DELETE/.	EDIT 359
C	LEN	WIDTH OF PAPER, FOR PRINTING CITATIONS. (128)	EDIT 360
C	LOCPAT(100)	LOCPAT(J) IS THE CHARACTER POSITION (LESS 1) IN	EDIT 361
C		PATREP OF FIRST CHARACTER IN J-TH PATTERN.	EDIT 362
C	LOCPT(100)	SUBSET OF LOCPAT APPLYING TO CURRENT FIELD OF CURRENT	EDIT 363
C		CITATION.	EDIT 364
C	LOCREP(100)	LOCREP(J) IS THE CHARACTER POSITION (LESS 1) IN	EDIT 365
C		PATREP OF FIRST CHARACTER IN J-TH REPLACEMENT.	EDIT 366
C	LOCRP(100)	SUBSET OF LOCREP APPLYING TO CURRENT FIELD OF CURRENT	EDIT 367
C		CITATION.	EDIT 368
C	LPRINT(100)	CITATION LIST IN /PRINT/.	EDIT 369
C	LREP(2,100)	THE J-TH REPLACEMENT APPLIES TO CITS LREP(1,J) THRU	EDIT 370
		LREP(2,J) INCLUSIVE.	EDIT 371
C	MAYBE(64)	MAYBE(J)=1 IF CHARACTER CODE J-1 IS THE FIRST	EDIT 372
C		CHARACTER OF ANY ONE OF THE NRP PATTERNS IN CURRENT	EDIT 373
		SUBSET, MAYBE(J)=0 OTHERWISE.	EDIT 374
	NTX	INPUT TAPE NUMBER.	EDIT 375
C	NTY	OUTPUT TAPE NUMBER.	EDIT 376
C	MXDEL	MAX. NO. OF ENTRIES IN LDEL ARRAY.	EDIT 377
C	NCHPAT(100)	NCHPAT(J) IS THE NUMBER OF CHARACTERS IN PATTERN J.	EDIT 378
C	NCHPT(100)	SUBSET OF NCHPAT APPLYING TO CURRENT FIELD OF CURRENT	EDIT 379
C		CITATION.	EDIT 380
C	NCHREP(100)	NCHREP(J) IS THE NUMBER OF CHARACTERS IN REPLACEMENT	EDIT 381
C		J.	EDIT 382
C	NCHRP(100)	SUBSET OF NCHREP APPLYING TO CURRENT FIELD OF CURRENT	EDIT 383
C		CITATION.	EDIT 384
C	NDEL	LENGTH OF LDEL ARRAY.	EDIT 385
C	NFLD	NUMBER OF FLDX LABELS.	EDIT 386
C	NPRINT	LENGTH OF LPRINT ARRAY.	EDIT 387
C	NREP	NUMBER OF REPLACEMENTS IN /REPLACE/.	EDIT 388
C	NRP	NUMBER OF REPLACEMENTS APPLYING TO CURRENT FIELD OF	EDIT 389
C		CURRENT CITATION.	EDIT 390
C	ORIG	TRUE IFF /ORIGINAL/ APPEARS.	EDIT 391
C	PATREP(800)	CONTAINS ALL PATTERN-REPLACEMENT CHARACTERS FROM	EDIT 392
C		/REPLACE/.	EDIT 393
C	PRIN	TRUE IFF CURRENT OUTPUT TAPE CITATION IS TO BE	EDIT 394
C		PRINTED.	EDIT 395
C	TINK	TRUE IFF CURRENT CITATION IS IN THE SCOPE OF A	EDIT 396
C		REPLACEMENT.	EDIT 397
	END		EDIT 398

PROGRAM GALLEY(INPUT,OUTPUT,TAPE1,	GALLEY 2
, TAPE50=INPUT,TAPE51=OUTPUT)	GALLEY 3
C GALLEY PROGRAM, PART OF THE FAMULUS SYSTEM,	GALLEY 4
COMMON /CORE/NCHW,BLANKS,BLANK	GALLEY 5
INTEGER BLANKS,BLANK	GALLEY 6
COMMON /FAM/SLASH,STAR,COMMA,ZERO,MXCHAR,MXCPES,IXWDES,MXCHW,	GALLEY 7
MXFREE,MXWFRE	GALLEY 8
INTEGER SLASH,STAR,COMMA,ZERO	GALLEY 9
COMMON /FIEL/FIELD(10),IFIELD(10),NFIELD,KDESCR	GALLEY10
INTEGER FIELD	GALLEY11
COMMON /INPT/CARD(10),KOLUM,HOLD,ENDCAR,HLACAR,CONCAR,ERRIN,NWC,	GALLEY12
MXFL,MAXCAR,NCARIN,KHR,KIN,LIT,LITX,SYM(5),CONR,NCSYM,CATLOG,	GALLEY13
KARA,KARB	GALLEY14
INTEGER CARD,SYM	GALLEY15
LOGICAL HOLD,ENDCAR,BLACAR,CONCAR,ERRIN,CATLOG	GALLEY16
COMMON /PAGING/IPAGE,LINES,MAXLIN,PAGHD(13)	GALLEY17
INTEGER PAGHD	GALLEY18
COMMON /TEST/ITEST	GALLEY19
INTEGER CIT(540),ID(13),IDA(13),LSEL(100),LIST(3)	GALLEY20
INTEGER JLAB(10)	07/18/43
LOGICAL MEMBER,FCHOI,NONO	07/18/44
ATA MXWID,MXWID,MXSEL / 20,128,100 /	07/18/45
C INITIALIZATION,	GALLEY23
CALL USHER(IDA,KERR)	GALLEY24
KE=1	GALLEY25
IF(KERR.NE.0) GO TO 900	GALLEY26
LCIT=0	GALLEY27
MTIN=1	GALLEY28
CALL REWIND(MTIN)	GALLEY29
LATE=100000	GALLEY30
LSEL(1)=1	GALLEY31
LSEL(2)=-LATE	GALLEY32
LSEL(3)=0	GALLEY33
WIDTH=MXWID	GALLEY34
KUBE=0	GALLEY35
FCHOI=.FALSE,	07/16/46
NONO=.FALSE,	07/18/47
C PROCESS THE TAPE ID AND HEADER INFORMATION	07/18/48
CALL IDI(0,MTIN,IL,FIELD,IFIELD,NFIELD,KDESCR)	07/18/49
CALL IDPRIN(ID,FIELD,IFIELD,NFIELD,KDESCR,10,10H1,PUT TAPE)	07/16/50
CALL SIMILE(1,ID,0,MXFREE,IDA,0,MXFREE,LIKE)	07/18/51
NR=6	07/18/52
IF(LIKE.NE.1) GO TO 900	37/18/53
C NEXT STATEMENT,	GALLEY36
10 CALL SETSYM	GALLEY37
IF(ENDCAR) GO TO 40	GALLEY38
CALL FABRIC(KON)	GALLEY39
KE=2	GALLEY40
IF(ERRIN) GO TO 900	GALLEY41
IF(KON.NE.3) GO TO 20	GALLEY42
C /SELECT/ STATEMENT,	GALLEY43
CALL GETNUM(LSEL,1,MXSEL,N,KERR)	GALLEY44
KE=3	GALLEY45
IF(KERR.GT.0) GO TO 900	GALLEY46
LATE=IABS(LSEL(N))	GALLEY47
CALL KORAN(4HLSEL,LSEL,N+1)	GALLEY48
GO TO 10	GALLEY49

20 IF(KON,NE,15) GO TO 30	GALLEY50
C /WIDTH/ STATEMENT.	GALLEY51
CALL GETNUM(LIST,1,1,N,KERR)	GALLEY52
KE=4	GALLEY53
IF(KERR,GT,0) GO TO 900	GALLEY54
NWIDTH = MAX0(MIN0(LIST(1),MXWID),MNWID)	07/18/54
GO TO 10	GALLEY56
C /PRINT BY FIELDS/	GALLEY57
30 IF(KON,NE,19) GO TO 90	07/18/55
KUBE=1	GALLEY59
GO TO 10	GALLEY60
90 IF (KON ,NE, 21) GO TO 100	07/18/56
C /NO NUMBERS/ OPTION TO DELETE CITATION NUMBERS IN OUTPUT FROM PRETTY	07/18/57
NONO = .TRUE.	07/18/58
GO TO 10	07/18/59
C /FIELD CHOICE/	07/18/60
100 IF(KON,NE,20) GO TO 39	07/18/61
FCHOI=.TRUE.	07/18/62
CALL GETLAB(JLAB,1,NFIELD,NCHOI,KERR)	07/18/63
IF(KERR,GT,0) GO TO 900	07/18/64
DO 105 K=1,NCHOI	07/18/65
L=LOOK(FIELD,NFIELD,JLAB(K))	07/18/66
IF(L,GT,NFIELD) 33 TO 900	07/18/67
JLAB(K)=L	07/18/68
105 CONTINUE	07/15/69
GO TO 13	07/18/70
C EXTRANEOUS STATEMENT.	GALLEY61
39 KE=5	GALLEYh2
GO TO 900	GALLEY63
40 CALL PAGER(0)	07/18/71
C PROCESS ONE CITATION.	GALLEY71
50 CALL IOTAPE(0,MTIN,CIT,LAST)	GALLEY72
IF(LAST,EQ,1,OR,LCIT,GT,LATE) GO TO 60	GALLEY73
LCIT=LCIT+1	GALLEY74
IF(.(NOT, MEMBER(LSEL,MXSEL,LCIT)) GO TO 50	GALLEY75
JCIT = LCIT	07/18/72
IF (NONO) JCIT = 0	07/18/73
IF(KUBE,EQ,0) CALL PRETTY(CIT,JCIT,0,NWIDTH,FCHOI,JLAB,NCHOI)	07/18/74
IF (KUBE,EQ,1) CALL PRNCIT(CIT,LCIT,0,NWIDTH)	GALLEY77
GO TO 50	GALLEY78
C DONE.	GALLEY79
60 CALL PAGER(0)	GALLEY80
CALL REWIND(MTIN)	GALLEY81
CALL PAGER(1)	GALLEY82
PRINT 1000	GALLEY83
1000 FORMAT(23H PROCESSING IS COMPLETE)	GALLEY84
CALL PAGER(11)	GALLEY85
PRINT 8999	GALLEY86
8999 FORMAT(/	GALLEY67
1 /3(20X 17H XXXXX X X)	GALLEY88
2 /3(20X 17HX X X X)	GALLEY89
3 /3(20X 17YX X X X)	GALLEY90
4 /3(20X 17HX X X)	GALLEY91
5 /3(20X 17HX X X X)	GALLEY92
6 /3(20X 17HX X X X)	GALLEY93
7 /3(20X 17H XXXXX X X)	GALLEY94
8 //)	GALLEY95

STOP		GALLEY96
C ERRORS.		GALLEY97
900 IF (KOLUM,NE,1) CALL DIVEST		GALLEY98
CALL PAGER(2)		GALLEY99
GO TO (901,901,901,901,901, 903),KE		GALLE100
901 PRINT 9001		GALLE101
9001 FORMAT(12H INPUT ERROR,116(1H*))		GALLE102
GO TO 950		GALLE103
903 PHINT 9003		GALLE104
9003 FORMAT(44H /ID/ TEXT DOES NOT MATCH ID FROM INPUT TAPE,84(1H*))		GALLE105
95; PRINT 9050,KE		GALLE1 a
9050 FORMAT(118X,6HGALLEY I4)		GALLE107
STOP		GALLE108
C CIT(450) CURRENT CITATION.		GALLE109
C LCIT CITATION NUMBER.		GALLE110
C IDA(13) TEXT FROM /ID/.		GALLE111
C LSEL(100) LIST FFOY /SELECT/.		GALLE112
C NWIDTH LINE WIDTH FROM /WIDTH/.		GALLE113
C ID(13) ID FROM INPUT TAPE PREFACE RECORD.		GALLE114
C ITIN INPUT TAPE.		GALLE115
C MXSEL MAXIMUM LENGTH OF LSEL ARRAY.		GALLE116
C MAXWD MAXIMUM POSSIBLE NWIDTH.		GALLE117
C LIST(3) TEMPORARY STORAGE.		GALLE118
C LATE LARGST CITATION NUMBER IN LSEL.		GALLE119
END		GALLE120

PROGRAM INDEX (INPUT,OUTPUT,PUNCH=65,TAPE1,	INDEX 2
, TAPE50=INPUT, TAPE51=OUTPUT)	INDEX 3
C INDEX, PART OF THE FAMULUS SYSTEM.	INDEX 4
C PROGRAMMED BY ROBERT M. RUSSELL, JUNE 1968,	INDEX 5
COMMON /CORE/NCHW,BLANKS,BLANK	INDEX 6
INTEGER BLANKS,BLANK	INDEX 7
COMMON /FAM/SLASH,STAR,COMMA,ZERO,MXCHAR,MXCODES,MXWDES,MXCHW,	INDEX 8
MXFREE,MXWFRE	INDEX 9
INTEGER SLASH,STAR,COMMA,ZERO	INDEX 10
COMMON /FIEL/FIELD(10),IFIELD(10),NFIELD,KDESCR	INDEX 11
INTEGER FIELD	INDEX 12
COMMON /INPT/CARD(10),KOLUM,HOLD,ENDCAR,BLACAR,CONCAR,ERRIN,NWC,	INDEX 13
MXFL,MAXCAR,NCARIN,KHR,KIN,LIT,LITX,SYM(5),NUMB,NCSYM,CATLOG,	INDEX 14
KARA,KARB	INDEX 15
INTEGER CARD,SYM	INDEX 16
LOGICAL HOLD,ENDCAR,BLACAR,CONCAR,ERRIN,CATLOG	INDEX 17
COMMON /PAGING/PAGE,LINES,MAXLIN,PAGHED(13)	INDEX 18
INTEGER PAGHED, SEMIC	INDEX 19
COMMON /TEST/ITEST	INDEX 20
C	INDEX 21
COMMON CIT(3,180), DICT(6500),LDICT(4001), HASH(9511),	INDEX 22
+ LOCCIT(8000), LOCDES(8000)	INDEX 23
INTEGER CIT,DICT,HASH	INDEX 24
C	INDEX 25
INTEGER ID(13),WORD(5),LINE(6),LSEL(100),IDA(13)	INDEX 26
EQUIVALENCE (KRANK,HASH)	INDEX 27
INTEGER KRANK(8000)	INDEX 28
LOGICAL MEMBER,PUNCHY	INDEX 29
C INITIALIZE,	INDEX 33
CALL USHER(IDA,KERR)	INDEX 31
C---DEFINE SEMICOLON AS DELIMITER CODE	RJO 2
SEMIC = KHAR(1H;,1)	RJC 3
KR=1	INDEX 32
IF(KERR.GT.0) GO TO 900	INDEX 33
MTIN=1	INDEX 34
CALL REWAND(MTIN)	INDEX 35
C-----	
MXENT=4000	INDEX 36
MXDICT=6500*NCHW	INDEX 37
MXHASH=9511	INDEX 38
C-----	
MXSEL=100	INDEX 39
NDICT=0	INDEX 40
LDICT(1)=0	INDEX 41
LATE=100000	INDEX 42
LSEL(1)=1	INDEX 43
LSEL(2)=-LATE	INDEX 44
LSEL(3)=0	INDEX 45
CALL MOVE(0,HASH,-MXHASH)	INDEX 46
MXLOC=8000	INDEX 47
LCT=0	INDEX 48
NLOC=0	INDEX 49
NLC=0	INDEX 50
NDCT=0	INDEX 51
LDCT=0	INDEX 52
PUNCHY=.FALSE.	INDEX 53
C NEXT STATEMENT,	INDEX 54

10 KE=2	INDEX 55
IF(ERRIN) GO TO 900	INDEX 56
IF(ITEST,NE,0) CALL PAGER(-4)	INDEX 57
CALL SETSYM	INDEX 58
IF(ENDCAR) GO TO 50	INDEX 59
CALL FABRIC(KON)	INDEX 60
KE=3	INDEX 61
IF(ERRIN) 60 TO 900	INDEX 62
IF(KOLUM,NE,1) CALL DIVEST	INDEX 63
IF(KON,NE,3) GO TO 20	INDEX 64
C /SELECT/ STATEMENT,	INDEX 65
CALL GETNUM(LSEL,1,MXSEL,N,KERR)	INDEX 66
KE=4	INDEX 67
IF(KERR,GT,0) GO TO 900	INDEX 68
LATE=IABS(LSEL(N))	INDEX 69
CALL KORAN(4HLSEL,LSEL,N+1)	INDEX 70
GO TO 10	INDEX 71
20 IF(KON,NE,17) GO TO 30	INDEX 72
C /PUNCH/ STATEMENT,	INDEX 73
PUNCHY=.TRUE.	INDEX 74
GO TO 10	INDEX 75
C EXTRANEOUS STATEMENT,	INDEX 76
30 CONTINUE	INDEX 77
KE=5	INDEX 78
GO TO 900	INDEX 79
C FINISHED WITH CARD INPUT,	INDEX 80
50 CALL PAGER(-4)	INDEX 81
C READ PREFACE FROM INPUT TAPE AND CHECK IT,	INDEX 82
CALL IOID(0,MTIN,ID,FIELD,IFIELD,NFIELD,KDESCR)	INDEX 83
CALL IDPRIN(ID,FIELD,IFIELD,NFIELD,KDESCR,10,10HINPUT TAPE)	INDEX 84
CALL SIMILE(1,ID,0,MXFREE,IDA,0,MXFREE,LIKE)	INDEX 85
KE=6	INDEX 86
IF(LIKE,NE,1) GO TO 900	INDEX 87
KE=7	INDEX 88
IF(KDESCR,LT,1,OR,KDESCR,GT,NFIELD) GO TO 900	INDEX 89
LCIT=0	INDEX 90
C READ NEXT CITATION FROM INPUT TAPE,	INDEX 91
100 IF(LCIT,GE,LATE) GO TO 200	INDEX 92
IF(ITEST,NE,0) CALL PAGER(-10)	INDEX 93
CALL KORAN(4HLCIT,LCIT+1,1)	INDEX 94
CALL IOTAPE(0,MTIN,CIT,LAST)	INDEX 95
IF(LAST,EQ,1) GO TO 200	INDEX 96
LCIT=LCIT+1	INDEX 97
IF(.NOT.MEMBER(LSEL,MXSEL,LCIT)) GO TO 100	INDEX 98
C PREPARE TO SCAN DESCRIPTOR FIELD,	INDEX 99
NC=CIT(1,KDESCR)	INDEX 100
LC=CIT(2,KDESCR)	INDEX 101
IF(NC,LE,0) GO TO 140	INDEX 102
C PICK UP NEXT DESCRIPTOR IN FIELD,	INDEX 103
110 CALL SCOUT(0,0,BLANK,CIT,LC,NC,M)	INDEX 104
LC=LC+M	INDEX 105
NC=NC-M	INDEX 106
C---CHANGE DELIMITER IN DESCRIPTOR FIELD TO SEMICOLON - RJ OLSON 07/73	RJ0 4
CALL SCOUT(0,1,SEMIC,CIT,LC,NC,NN)	HJC
CALL SCOUT(1,0,BLANK,CIT,LC,NN,M)	INDEX 108
N=MIN0(NN-M,MXCDER)	INDEX 109
IF(N,LE,0) GO TO 130	INDEX 110

CALL TALMUD(6HCITDES,CIT,LC,N)	INDEX111
CALL SCAT(CIT,LC,N,DICT,LDICT,NDICT,HASH,MXHASH,JDICT,JHASH)	INDEX112
IF(JDICT,LE,NDICT) GO TO 120	INDEX113
C DESCRIPTOR IS NEW, PUT IT IN DICT,	INDEX114
NDICT=NDICT+1	INDEX115
IF(NDICT,GT,MXENT) GO TO 150	INDEX116
LD=LDICT(NDICT)	INDEX117
LDICT(NDICT+1)=LD+N	INDEX118
IF(LD+N,GT,MXDICT) GO TO 155	INDEX119
CALL CHOP(CIT,LC,DICT,LD,N)	INDEX120
HASH(JHASH)=NDICT	INDEX121
CALL TALMUD(6HNEWDES,DICT,LD,N)	INDEX122
C ENTER DESCRIPTOR IN LOCCIT AND LOCDES,	INDEX123
120 IF(NLOC,GE,MXLOC) GO TO 155	INDEX124
NLOC=NLOC+1	INDEX125
LOCCIT(NLOC)=LCIT	INDEX126
LOCDES(NLOC)=JDICT	INDEX127
CALL KORAN(6HLOCCIT,LOCCIT,NLOC)	INDEX128
CALL KORAN(6HLOCDES,LOCDES,NLOC)	INDEX129
C MOVE TO NEXT DESCRIPTOR IN FIELD,	INDEX130
130 NN=MIN0(NN+1,NC)	INDEX131
LC=LC+NN	INDEX132
NC=NC-NN	INDEX133
IF(NC,GT,0) GO TO 110	INDEX134
C FINISHED SCANNING CITATION,	INDEX135
140 LCT=LCIT	INDEX136
NLC=NLOC	INDEX137
NDCT=NDICT	INDEX138
LDCT=LDICT(NDICT+1)	INDEX139
GO TO 100	INDEX140
C FULL LDICT,	INDEX141
150 NDICT=NDICT-1	INDEX142
C FULL DICT,	INDEX143
C FULL LOCCIT AND LOCDES,	INDEX144
155 CALL PAGER(7)	INDEX145
PRINT 1000 , LCIT , LCT	INDEX146
1000 FORMAT(3 (60(" *")/) / " IN CITATION",16," , THE CAPACITY"	INDEX147
+, " OF THE PROGRAM IS EXCEEDED,"/" PRINTOUT WILL INCLUDE ALL DESC"	INDEX148
+, "RIPTORS AND CITATIOYS THROUGH CITATION NUMBER",16," ,"/60(" *")/)	INDEX149
1500 FORMAT (40X,"STATISTICS"/	
. 39X,7HCURRENT,3X,7HMAXIMUM/	INDEX151
. 22H NUMBER OF DESCRIPTORS 124,110/	INDEX152
. 40H NUMBER OF CHARACTERS IN ALL DESCRIPTORS 16,110/	INDEX153
. 38H NUMBER OF ENTRIES IN DESCRIPTOR INDEX 18,110)	INDEX154
NLOC=NLC	INDEX155
C FINISHED COLLECTING, NOW SORT,	INDEX156
200 CALL REWAND(MTIN)	INDEX157
CALL PAGER(5)	
PRINT 1500 , NDCT,MXENT,LDCT,MXDICT,NLC,MXLOC	INDEX
CALL PAGER(0)	INDEX158
IF(NLOC,LE,0) CALL HALT	INDEX159
CALL KORAN(6HLOCCIT,LOCCIT,NLOC)	INDEX160
CALL KORAN(6HLOCDES,LOCDES,NLOC)	INDEX161
CALL SHELL1(LOCDES,KRANK,NLOC)	INDEX162
CALL KORAN(5HKRANK,KRANK,NLOC)	INDEX163
DO 202 J=1,NLOC	INDEX164
KR=KRANK(J)	INDEX165

L=LOCDES(JR)	INDEX166
LD=LDICT(L)	INDEX167
LD=LDICT(L+1)-LD	INDEX168
202 CALL TALMUD(4HDICT,LDICT,LD,NL)	INDEX169
CALL PAGER(0)	INDEX170
CALL PAGER(3)	INDEX171
PRINT 2000	INDEX172
2001 FORMAT(/6X,9HIN D E X/1X)	INDEX173
NWL=6	INDEX174
NCL=4	INDEX175
IF(ITEST,NE,0) NWL=ITEST	INDEX176
NCL=NWL*NCHE	INDEX177
CALL SPOT(NWL,NCL,LCIT,0,4HIIII,24H NL JC LCIT)	INDEX178
JB=0	INDEX179
C PRINT NEXT DESCRIPTOR,	INDEX180
210 JA=JB+1	INDEX181
DO 220 J=JA,NLOC	INDEX182
JR=KRANK(J)	INDEX183
K=LOCDES(JR)	INDEX184
IF(J,EQ,JA) KD=K	INDEX185
IF(K,NE,KD) GO TO 230	INDEX186
220 CONTINUE	INDEX187
J=NLOC+1	INDEX188
230 JB=J-1	INDEX189
IF(LINES.LT,2) CALL PAGER(0)	INDEX190
LD=LDICT(KD)	INDEX191
ND=LDICT(KD+1)-LD	INDEX192
CALL SPOT(JA,JB,KD,0,4HIIII,24HJA J, KD	INDEX193
CALL MOVE(BLANKS,WORD,-MXWDES)	INDEX194
CALL CHOP(DICT,LD,WORD,0,ND)	INDEX195
CALL PAGER(1)	INDEX196
PRINT 3000,(WORD(I),I=1,MXWDES)	INDEX197
IF(PUNCHY) PUNCH 3000,(WORD(I),I=1,MXWDES)	INDEX198
3000 FORMAT(1X,6A8)	INDEX199
C PRINT CITATION NUMBERS FOR CURRENT DESCRIPTOR,	INDEX200
CALL MOVE(BLANKS,LINE,-NWL)	INDEX201
DO 280 J=JA,JB	INDEX202
JR=KRANK(J)	INDEX203
KC=LOCCT(JR)	INDEX204
IF(J,EQ,JA) GO TO 260	INDEX205
IF(NL,GT,0) GO TO 270	INDEX206
250 CALL PAGER(1)	INDEX207
PRINT 4000,(LINE(I),I=1,NWL)	INDEX208
IF(PUNCHY) PUNCH 4000,(LINE(I),I=1,NWL)	INDEX209
4000 FORMAT(5X,6A8)	INDEX210
CALL MOVE(BLANKS,LINE,-NWL)	INDEX211
260 LL=0	INDEX212
NL=NCL	INDEX213
270 CONTINUE	INDEX214
C CALL SPOT(J,JR,KC,NL,4HIIII,24HJ JR KC NL)	INDEX215
CALL CIPHER(KC,LINE,LL,NL)	INDEX216
CALL TALMUD(4HLINE,LINE,0,LL)	INDEX217
IF(NL,LE,0) GO TO 250	INDEX218
280 CONTINUE	INDEX219
CALL PAGER(1)	INDEX220
PRINT 4000,(LINE(I),I=1,NWL)	INDEX221
IF(PUNCHY) PUNCH 4000,(LINE(I),I=1,NWL)	INDEX222

IF (JB.LT.NLOC) GO TO 210	INDEX223	
2 DONE,	INDEX224	
PRINT 8999	INDEX225	
8999 FORMAT(16H1F I N I S H E D)	INDEX226	
STOP	INDEX227	
C ERRORS,	INDEX228	
900 IF (KOLUM.GT.1) CALL DVEST	INDEX229	
CALL PAGER(2)	INDEX230	
GO TO (901,901,901,901,902, 903,904),KE	INDEX231	
901 PRINT 9001	INDEX232	
9001 FORMAT(12H INPUT ERROR,116(1H*))	INDEX233	
GO TO 950	INDEX234	
902 PRINT 9007	INDEX235	
9002 FORMAT(21H EXTRANEOUS STATEMENT,107(1H*))	INDEX236	
GO TO 950	INDEX237	
903 PRINT 9003	INDEX238	
9003 FORMAT(47H ID ON INPUT TAPE DOES NOT MATCH /ID/ STATEMENT,	INDEX239	
,81(1H*))	INDEX240	
GO TO 950	INDEX241	
904 PRINT 9004	INDEX242	
9004 FORMAT(57H NO DESCRIPTOR FIELD HAS BEEN DEFINED FOR THIS INPUT TAPE	INDEX243	
,E, 71(1H*))	INDEX244	
950 PRINT 9050,KE	INDEX245	
9050 FORMAT(118X,6HINDEX I4)	INDEX246	
STOP	INDEX247	
C CIT(,)	CURRENT CITATION,	INDEX248
C DICT(,)	TEXT OF DESCRIPTORS,	INDEX249
C LDICT(,)	POINTERS TO BEGINNING OF EACH DESCRIPTOR IN DICT,	INDEX250
C HASH(,)	HASH CODE TABLE, FOG HASH CODE K, HASH(K)=0 MEANS ENTRY	INDEX251
C	IS NOT IN DICT, WHILE HASH(K)=L=NONZERO MEANS ENTRY IS IN	INDEX252
C	DICT AND STARTS AT LDICT(L),	INDEX253
C LOCCIT(,)	CITATION NUMBERS CORRESPONDING TO DESCRIPTOR NUMBERS IN	INDEX254
C	LOCDES,	INDEX255
C LOCDES(,)	DESCRIPTOR NUMBERS, LET LOCDES(K)=J AND LET LOCCIT(K)=L,	INDEX256
C	THIS MEANS THAT DESCRIPTOR NUMBER J (LOCATED IN DICT AT	INDEX257
C	LDICT(J)) HAS BEEN FOUND IN CITATION L,	INDEX258
C WORD(,)	CURRENT DESCRIPTOR,	INDEX259
C KRANK(,)	INDICATES ALPHABETICAL ORDERING OF LOCDES ENTRIES,	INDEX260
C IDA(,)	ID FROM /ID/ STATEMENT,	INDEX261
C ID(,)	ID FROM INPUT TAPE,	INDEX262
C LATE	LAST CITATION THAT NEEDS TO BE INSPECTED,	INDEX263
C LCIT	CURRENT CITATION NUMBER,	INDEX264
C LSEL(,)	CITATION LIST FROM /SELECT/ STATEMENT,	INDEX265
C MTIN	INPUT TAPE,	INDEX266
C MXDICT	MAXIMUM NUMBER OF CHARACTERS IN DICT,	INDEX267
C MXENT	MAXIMUM NUMBER OF ENTRIES IN DICT,	INDEX268
C MXHASH	LENGTH OF HASH TABLE,	INDEX269
C MXSEL	MAXIMUM LENGTH OF LSEL,	INDEX270
C NDICT	NUMBER OF ENTRIES IN DICT,	INDEX271
C NLOC	NUMBER OF ENTRIES IN LOCCIT AND LOCDES,	INDEX272
C MXLOC	MAXIMUM LENGTH OF LOCCIT AND LOCDES,	INDEX273
C LCT	LAST CITATION INSPECTED,	INDEX274
END	INDEX275	

PROGRAM MERGE(INPUT,OUTPUT,TAPE1,TAPE2,TAPE3,	MERGE 1
, TAPE5=INPUT,TAPE51=OUTPUT)	MERGE 2
C MERGE PROGRAM, PART OF THE FAMULUS SYSTEM.	MERGE 3
C PROGRAMMED BY ROBERT A. RUSSELL.	MERGE 4
COMMON /CORE/MCHW,BLANKS,BLANK	MERGE 5
INTEGER BLANKS,BLANK	MERGE 6
COMMON /FAM/SLASH,STAR,COMMA,ZERO,MXCHAR,MXCODES,MXWDES,MXCHW,	MERGE 7
, MXFREE,MXWFE	MERGE 8
INTEGER SLASH,STAR,COMMA,ZERO	MERGE 9
COMMON /FIEL/FIELD(10),IFIELD(10),VFIELD,KDESCR	MERGE 10
INTEGER FIELD	MERGE 11
COMMON /INPT/CARD(10),KOLUM,HOLD,ENDCAR,BLACAR,CONCAR,ERRIN,NWC,	MERGE 12
, MXFL,MAXCAR,NCARIN,KHR,KIN,LIT,LITX,SYM(5),NUM,NCSYM,CATLOG,	MERGE 13
, KARA,KARB	MERGE 14
INTEGER IARD,SYM	MERGE 15
LOGICAL HOLD,ENDCAR,BLACAR,CONCAR,ERRIN,CATLOG	MERGE 16
COMMON /PAGING/IPAGE,LINES,MAXLIN,PAGHED(13)	MERGE 17
INTEGER PAGHED	MERGE 18
COMMON /TEST/ITEST	MERGE 19
INTEGER CITA(540),CITB(540),CITC(540),CIT(540)	MERGE 20
INTEGER IDA(13),IDB(13),IDC(13),IE(13)	MERGE 21
INTEGER LIST(3),IND(10),IFLD(10),CITX(540),CTX(3,180),CTE(3,180)	MERGE 22
EQUIVALENCE (CTB,CITB),(CTX,CITX)	MERGE 23
LOGICAL PRIN,STRATE	MERGE 24
DATA MXWC/540/	MERGE 25
C INITIALIZATION,	MERGE 26
CALL USHER(IDA,KERR)	MERGE 27
KE=1	MERGE 28
IF(KERR,IE,0) GO TO 900	MERGE 29
CALL MOVE(0,CITA,-MXWC)	MERGE 30
CALL MOVE(0,CITB,-MXWC)	MERGE 31
CALL MOVE(0,CITC,-MXWC)	MERGE 32
CALL MOVE(0,CIT,-MXWC)	MERGE 33
CALL MOVE(0,CITX,-MXWC)	MERGE 34
MTA=1	MERGE 35
MTB=2	MERGE 36
MTC=3	MERGE 37
CALL REWAND(MTA)	MERGE 38
CALL REWAND(MTB)	MERGE 39
CALL REWAND(MTC)	MERGE 40
LCITC=0	MERGE 41
LASTA=-1	MERGE 42
LASTB=-1	MERGE 43
MXWID=128	MERGE 44
WIDTH=MXWID	MERGE 45
PRIN=.FALSE.	MERGE 46
STRATE=.TRUE.	MERGE 47
CALL CHOP(IDA,0,IDC,0,MXFREE)	MERGE 48
C SECOND /ID/ STATEMENT, FOR ADDITIONS TAPE (MTB),	MERGE 49
CALL SETSYM	MERGE 50
KI=3	MERGE 51
IF(ENDCAR) GO TO 900	MERGE 52
CALL FABRIC(KON)	MERGE 53
KE=4	MERGE 54
IF(ERRIN.OR,KON,NE,1) GO TO 900	MERGE 55
CALL GOSSIP(IDB)	MERGE 56
KE=2	MERGE 57
	MERGE 58

IF(ERRIN) GO TO 900	MERGE 59
C /NEW II/ STATEMENT, FOR OUTPUT TAPE (MTC),	MERGE 60
CALL SETSYM	MERGE 61
IF(ENDCAR) GO TO 20	MERGE 62
CALL FABRIC(KON)	MERGE 63
IF(KON.EQ.5) GO TO 10	MERGE 64
KE=5	MERGE 65
IF(ERRIN.OR.KON.NE.4) GO TO 900	MERGE 66
CALL GOSSIP(IDC)	MERGE 67
KE=12	MERGE 68
IF(ERRIN) GO TO 900	MERGE 69
CALL SETSYM	MERGE 70
IF(ENDCAR) GO TO 20	MERGE 71
CALL FABRIC(KON)	MERGE 72
C /PRINT/ STATEMENT, IF ANY,	MERGE 73
10 CONTINUE	MERGE 74
KE=11	MERGE 75
IF(ERRIN.OR.KON.NE.5) GO TO 900	MERGE 76
PRIN=.TRUE,	MERGE 77
IF(ENDCAR) GO TO 20	MERGE 78
"ALL GETRUM(LIST,1,1,N,KERR)	MERGE 79
KE=13	MERGE 80
IF(KERR.GT.0) GO TO 900	MERGE 81
NWIDTH=LIST(1)	MERGE 82
CALL KORAN(6HNWIDTH,LIST,N+1)	MERGE 83
CALL KORAN(6HNWIDTH,NWIDTH,1)	MERGE 84
KE=10	MERGE 85
IF(.NOT.ENDCAR) GO TO 900	MERGE 86
20 CALL CHOP(IDC,0,PAGHD,0,MXFREE)	MERGE 87
CALL DIVEST	MERGE 88
C PREFACE RECORDS,	MERGE 89
CALL IOID(0,MTB,ID,FIELD,IFIELD,NFIELD,KDESCR)	MERGE 90
CALL IDPRIN(ID,FIELD,IFIELD,NFIELD,KDESCR,14,14HADDITIONS TAPE)	MERGE 91
CALL SIMILE(1,ID,0,MXFREE,IDB,0,MXFREE,LIKE)	MERGE 92
KE=6	MERGE 93
IF(LIKE.NE.1) GO TO 900	MERGE 94
NFLD=NFIELD	MERGE 95
CALL MOVE(IFIELD,IFLD,NFIELD)	MERGE 96
CALL IOII(0,MTA,ID,FIELD,IFIELD,NFIELD,KDESCR)	MERGE 97
CALL IDPRIN(ID,FIELD,IFIELD,NFIELD,KDESCR,11,11HMASTER TAPE)	MERGE 98
CALL SIMILE(1,ID,0,MXFREE,IDA,0,MXFREE,LIKE)	MERGE 99
KE=7	MERGE100
IF(LIKE.NE.1) GO TO 900	MERGE101
KE=8	MERGE102
IF(NFLD.NE.NFIELD) GO TO 900	MERGE103
CALL CHOP(IDC,0,IL,0,MXFREE)	MERGE104
CALL IOII(1,MTC,ID,FIELD,IFIELD,NFIELD,KDESCR)	MERGE105
CALL IDPRIN(ID,FIELD,IFIELD,NFIELD,KDESCR,11,11HOUTPUT TAPE)	MERGE106
DO 30 K=1,NFIELD	MERGE107
IF(IFIELD(K).NE.IFLD(K)) STRATE=.FALSE,	MERGE108
J=IFIELD(K)	MERGE109
30 IND(J)=IFLD(K)	MERGE110
CALL KORAN(3HIND,IND,NFIELD)	MERGE111
LCITA=0	MERGE112
LCITB=0	MERGE113
C NEXT CITATION,	MERGE114
50 IF(LASTA -1) GO TO 60	MERGE115

CALL IOTAPE(0,MTA,CITA,LASTA)	MERGE116
IF(LASTA.EQ.0) LCITA=LCITA+1	MERGE117
60 IF(LASTB.EQ.-1) GO TO 70	MERGE118
IF(STRATE) GO TO 64	MERGE119
CALL IOTAPE(0,MTB,CITX,LASTB)	MERGE120
IF(LASTB.EQ.0) GO TO 66	MERGE121
C REARRANGE AC TIONS CITATION SO ORDER OF FIELDS IS SAME AS MASTER TAPE	MERGE122
LB=3*NFIELD	MERGE123
62 K=1,NFIELD	MERGE124
KF=IND(K)	MERGE125
CTX(1,KF)	MERGE126
CTB(1,K)=	MERGE127
CTB(2,K)=LB*NCHW	MERGE128
CTB(3,K)=CTX(3,KF)	MERGE129
IF(N.EQ.0) GO TO 62	MERGE130
YF=(N+NCHW-1)/NCHW	MERGE131
LX=(CTX(2,YF)+NCHW-1)/NCHW	MERGE132
CALL MOVE(CITX(LX+1),CITB(LB+1),N)	MERGE133
LB=LB+N	MERGE134
62 CONTINUE	MERGE135
GO TO 66	MERGE136
64 CALL IOTAPE(0,MTB,CITB,LASTB)	MERGE137
66 IF(LASTB.EQ.0) LCITB=LCITB+1	MERGE138
70 IF(LASTA.EQ.1.AND.LASTB.EQ.1) GO TO 120	MERGE139
IF(LASTA.EQ.1) GO TO 90	MERGE140
IF(LASTB.EQ.1) GO TO 80	MERGE141
IF(LEADER(CITA,CITB)) 80,80,90	MERGE142
80 CALL COPCIT(CITA,CIT,0)	MERGE143
LASTA=-1	MERGE144
LCITX=LCITA	MERGE145
GO TO 100	MERGE146
90 CALL COPCIT(CITB,CIT,0)	MERGE147
LASTB=-1	MERGE148
LCITB=LCITB	MERGE149
100 IF(LCITC.EQ.0) CALL COPCIT(CIT,CITC,0)	MERGE150
N=9	MERGE151
IF(LEADER(CIT,CITC).EQ.-1) GO TO 900	MERGE152
CALL COPCIT(CIT,CITC,0)	MERGE153
LCITC=LCITC+1	MERGE154
CALL IOTAPE(1,MTC,CITC,0)	MERGE155
LCITXX=LCITX	MERGE156
IF(.NOT.FIN) GO TO 50	MERGE157
CALL PRETTY(CITC,LCITC,1,128)	MERGE158
GO TO 50	MERGE159
C DONE.	MERGE160
120 CONTINUE	MERGE161
CALL IOTAPE(1,MTC,CITC,1)	MERGE162
CALL REWAND(MTA)	MERGE163
CALL REWAND(MTB)	MERGE164
CALL REWAND(MTC)	MERGE165
CALL PAGER(-5)	MERGE166
CALL PAGER(1)	MERGE167
PRINT 1000,LCITC	MERGE168
1000 FORMAT(37H NUMBER OF CITATIONS ON OUTPUT TAPE =15)	MERGE169
CALL PAGER(-1)	MERGE170
CALL PAGER(1)	MERGE171
PRINT 2000	MERGE172

2000	FORMAT(21H PROCESSING COMPLETED)	MERGE173
	CALL PAGER(11)	MERGE174
	PRINT 8999	MERGE175
8999	FORMAT(/	MERGE176
1	/3(20x 17H XXXXX X X)	MERGE177
2	/3(20x 17HX X X X)	MERGE178
3	/3(20x 17HX X X X)	MERGE179
4	/3(20x 17HX X X)	MERGE180
5	/3(20x 17HX X X X)	MERGE181
6	/3(20x 17HX X X X)	MERGE182
7	/3(20x 17H XXYXX X X)	MERGE183
8	/)	MERGE184
	STOP	MERGE185
C	ERRORS.	MERGE186
900	IF(KOLUM,NE,1) CALL DIVEST	MERGE187
	CALL PAGER(2)	MERGE188
	GG TO (901,901,901,901,901,	MERGE189
	902,902,903,904,901,	MERGE190
	901,901,901),KE	MERGE191
901	PRINT 9001	MERGE192
9001	FORMAT(12H INPUT ERROR,116(1H*))	MERGE193
	GO TO 950	MERGE194
902	PRINT 9002	MERGE195
9002	FORMAT(37H /ID/ DOES NOT MATCH ID On INPUT TAPE,91(1H*))	MERGE196
	GO TO 950	MERGE197
903	PRINT 9003	MERGE198
9003	FORMAT(47H NUMBER OF FIELDS ON INPUT TAPES DOES NOT MATCH,81(1H*))	MERGE199
	GO TO 950	MERGE200
904	PRINT 9004	MERGE201
9004	FORMAT(27H CITATIONS ARE OUT OF ORDER,101(1H*))	MERGE202
950	PRINT 9050,KE	MERGE203
9050	FORMAT(118X,6HMERGE 14)	MERGE204
	IF(KE,NE,9) GO TO 960	MERGE205
	CALL PRNCIT(CITC,LCITXX,0,128)	MERGE206
	CALL PRNCIT(CIT,LCITX,0,128)	MERGE207
960	CONTINUE	MERGE208
	STOP	MERGE209
C	CITA(450) INPUT MASTER TAPE CITATION.	MERGE210
C	CITB(450) INPUT ADDITIONS TAPE CITATION.	MERGE211
C	CITC(450) OUTPUT TAPE CITATION.	MERGE212
C	LCITC NUMBER OF CITATIONS SO FAR WRITTEN OUT.	MERGE213
C	MTA MASTER TAPE.	MERGE214
C	MTB ADDITIONS TAPE.	MERGE215
C	MTC OUTPUT TAPE.	MERGE216
C	IDA(13) ID FROM FIRST /ID/ STATEMENT (MASTER).	MERGE217
C	IDB(13) ID FROM SECOND /ID/ STATEMENT (ADDITIONS).	MERGE218
C	IDC(13) ID FROM /NEW ID/ STATEMENT (OUTPUT).	MERGE219
C	ID(13) OUTPUT TAPE ID, ALSO USED AS TEMPORARY STORAGE FOR	MERGE220
C	ID FROM INPUT TAPES.	MERGE221
C	NFLD TEMPORARY STORAGE FOR NFIELD.	MERGE222
C	LASTA LASTA=-1, CITA IS WAITING FOR A CITATION FROM MTA,	MERGE223
C	LASTA=0, CITA CONTAINS A CITATION FROM MTA, LASTA=1,	MERGE224
C	MTA IS AT END OF FILE.	MERGE225
C	LASTB SIMILAR TO LASTA, BUT PERTAINS TO CITB AND MTB.	MERGE226
C	CIT(450) CITATION WAITING FOR CITC.	MERGE227
C	PRIN TRUE IFF /PRINT/ IS IN INPUT, CAUSING ALL CITATIONS	MERGE228
C	TO BE PRINTED WHILE MERGING.	MERGE229

C	NWIDTH	WIDTH OF PRINTED CITATION, WHEN PRI = TRUE.	MERGE230
C	MXWID	MAXIMUM NWIDTH.	MERGE231
C	STRATE	TRUE IFF REORDERING INDICATORS ARE THE SAME FOR BOTH	MERGE232
C		INPUT TAPES.	MERGE233
C	INI(10)	TELLS HOW TO REORDER CITH, WHEN STRATE = FALSE.	MERGE234
C	CITX(540)	TEMPORARY STORAGE OF CITH, WHEN STRATE = FALSE.	MERGE235
C	CIX(3,180)	SAME AS CITX(540).	MERGE236
C	CIR(3,180)	SAME AS CITX(540).	MERGE237
C	IFLD(10)	TEMPORARY STORAGE FOR IFIELD FROM ADDITIONS TAPE.	MERGE238
	END		MERGE239

PROGRAM OSSIFY(INPUT,OUTPUT,PUNCH,TAPE1,	OSSIFY 2
TAPE50=INPUT,TAPE51=OUTPUT)	OSSIFY 3
C OSSIFY,	OSSIFY 4
C PUNCHES A SELECTED SUBSET OF CITATIONS FROM A FILE, USUALLY FOR	OSSIFY 5
C SAFEKEEPING.	OSSIFY 6
COMMON /CORE/NCHW,BLANKS,BLANK	OSSIFY 7
INTEGER BLANKS,BLANK	OSSIFY 8
COMMON /FAM/SLASH,STAR,COMMA,ZERO,MXCHAR,MXCDIS,MXWDES,MXCHW,	OSSIFY 9
MXFREE,MXWFRE	OSSIFY10
INTEGER SLASH,STAR,COMMA,ZERO	OSSIFY11
COMMON /FIEL/FIELD(10),IFIELD(10),NFIELD,KDESCR	OSSIFY12
INTEGER FIELD	OSSIFY13
COMMON /INPT/CARD(10),KOLUM,HOLD,ENDCAR,BLACAR,CONCAR,ERRIN,NWC,	OSSIFY14
MXFL,MXCAR,NCARIN,KHR,KIN,LIT,LITX,SYM(5),NUMR,NCSYM,CATLOG,	OSSIFY15
KARA,KARB	OSSIFY16
INTEGER CARD,SYM	OSSIFY17
LOGICAL HOLD,ENDCAR,BLACAR,CONCAR,ERRIN,CATLOG	OSSIFY18
COMMON /TEST/ITEST	OSSIFY19
INTEGER CIT(3,180),LSEL(100),ID(13),IDD(13),KARD(10)	OSSIFY20
LOGICAL MEMBER	OSSIFY21
CALL USHER(IDD,KERR)	OSSIFY22
KE=1	OSSIFY23
IF(KERR.NE.0) GO TO 900	OSSIFY24
MT=1	OSSIFY75
CALL REWIND(MT)	OSSIFY26
MXSEL=100	OSSIFY27
CALL IOII(0,MT,ID,FIELD,IFIELD,NFIELD,KDESCR)	OSSIFY28
CALL IDPRIN(ID,FIELD,IFIELD,NFIELD,KDESCR,10,10HINPUT TAPE)	OSSIFY29
CALL SIMILE(1,ID,0,MXFREE,IDD,0,MXFREE,M)	OSSIFY30
KE=2	OSSIFY31
IF(M.NE.1) GO TO 900	OSSIFY32
LATE=100000	OSSIFY33
LSEL(1)=1	OSSIFY34
LSEL(2)=-LATE	OSSIFY35
LSEL(3)=0	OSSIFY36
IF(ENDCAR) GO TO 10	OSSIFY37
CALL SETSYM	OSSIFY38
CALL FABRIC(KON)	OSSIFY39
IF(ERRIN.OR.KON.NE.3) GO TO 900	OSSIFY40
C /SELECT/ STATEMENT.	OSSIFY41
CALL GETNUM(LSEL,1,MXSEL,NSEL,KERR)	OSSIFY42
KE=4	OSSIFY43
IF(KERR.EQ.1) GO TO 900	OSSIFY44
KE=5	OSSIFY45
IF(KERR.GE.2) GO TO 900	OSSIFY46
LATE=IABS(LSEL(NSEL))	OSSIFY47
CALL KORAN(6HSELECT,LSEL,NSEL+1)	OSSIFY48
10 NPUN=0	OSSIFY49
LCIT=0	OSSIFY50
C READ CITATION FROM INPUT TAPE,	OSSIFY51
40 LCIT=LCIT+1	OSSIFY52
CALL IOTAPE(0,MT,CIT,LEND)	OSSIFY53
IF(LEND.EQ.1) GO TO 90	OSSIFY54
IF(LCIT.GT.LATE) GO TO 90	OSSIFY55
IF(.NOT.MEMBER(LSEL,MXSEL,LCIT)) GO TO 40	OSSIFY56
IF(NPUN.GT.0) PUNCH 3000	OSSIFY57
	OSSIFY58

3000	FORMAT(1X)	OSSFY50
	DO 70 K=1, NFIELD	OSSFY51
	KF=FIELD(K)	OSSFY61
	NF=CIT(1,KF)	OSSFY62
	LF=CIT(2,KF)	OSSFY63
	CALL TALK (FIELD(KF), CIT, LF, NF)	OSSFY64
	IF(NF, LE, 0) GO TO 70	OSSFY65
50	NIB=MIN0(NF, 75)	OSSFY66
	CALL MOVE(BLANKS, KAR), -NWC)	OSSFY67
	IF(LF, EQ, CIT(2, KF)) KAR(J)=FIELD(KF)	OSSFY68
	CALL CHOP(CIT, LF, KAR, 5, NIB)	OSSFY69
	PUNCH 10001 (KAR(J), J=1, NWC)	OSSFY70
1000	FORMAT(11A8)	OSSFY71
	IF(ITEST, E 3.0) GO TO 50	OSSFY72
	CALL PAGER(1)	OSSFY73
	PRINT 2000, (KAR(J), J=1, NWC)	OSSFY74
2000	FORMAT(9H PUNCHING, 7(1H,), 10A8)	OSSFY75
60	LF=LF+NIB	OSSFY76
	NF=NF-NIB	OSSFY77
	IF(NF, GT, 0) GO TO 50	OSSFY78
70	CONTINUE	OSSFY79
	PUN=NPU,+1	OSSFY80
	GO TO 40	OSSFY81
C ALL ONE.		OSSFY82
90	CALL REWARD(MT)	OSSFY83
	CALL PAGER(-5)	OSSFY84
	CALL PAGER(1)	OSSFY85
	PRINT 4000, NPUN	OSSFY86
4000	FORMAT(20H PUNCHING COMPLETED, 16.19H CITATIONS PUNCHED,)	OSSFY87
	CALL PAGER(-5)	OSSFY88
	CALL PAGER(11)	OSSFY89
	PRINT 8999	OSSFY90
8999	FORMAT(/	OSSFY91
1	/3(20X 17H XXXXX X X)	OSSFY92
2	/3(20X 17HX X X X)	OSSFY93
3	/3(20X 17HX X X X)	OSSFY94
4	/3(20X 17HX X X)	OSSFY95
5	/3(20X 17HX X X X)	OSSFY96
6	/3(20X 17HX X X X)	OSSFY97
7	/3(20X 17H XXXXX X X)	OSSFY98
8	/)	OSSFY99
	STOP	OSSF100
C ERRORS.		OSSF101
900	IF(KOLUM, GT, 1) CALL DIVERST	OSSF102
	CALL PAGER(2)	OSSF103
	GO TO (901, 902, 903, 904, 905), KE	OSSF104
901	PRINT 9001	OSSF105
9001	FORMAT(32H THE FIRST STATEMENT IS NOT /I', 96(1H*))	OSSF106
	GO TO 950	OSSF107
902	PRINT 9002	OSSF108
9003	FORMAT(28H TAPE ID DOES NOT MATCH /I', 100(1H*))	OSSF109
	GO TO 950	OSSF110
903	PRINT 9003	OSSF111
9005	FORMAT(39H NOT LOOKING FOR THIS TYPE OF STATEMENT, 89(1H*))	OSSF112
	GO TO 951	OSSF113
904	PRINT 9004	OSSF114
9004	FORMAT(30H THE /SELECT/ LIST IS TOO LONG, 96(1H*))	OSSF115

GO TO 950	OSSF116
905 PRINT 9005	OSSF117
9005 FORMAT(23H ERROR IN /SELECT/ LIST,105(1H*))	OSSF118
950 PRINT 9050, KE	OSSF119
9050 FORMAT(11EX,6HOSIFY I4)	OSSF120
STOP	OSSF121
C FORTRAN VARIABLES...	OSSF122
C CIT(3,150) CURRENT CITATION.	OSSF123
C LCIT CITATION NUMBER.	OSSF124
C LSEL(100) LIST OF CITATIONS TO PUNCH, FROM /SELECT/.	OSSF125
C ID(15) ID OF INPUT TAPE.	OSSF126
C MT INPUT TAPE.	OSSF127
C IDD(15) ID FROM /ID/.	OSSF128
C KARD(10) CURRENT PUNCH CARD IMAGE.	OSSF129
C MXSEL MAXIMUM LENGTH OF LSEL ARRAY.	OSSF130
C NPUN NUMBER OF CITATIONS PUNCHED.	OSSF131
C NSEL NUMBER OF ENTRIES IN LSEL ARRAY.	OSSF132
C LATE LARGEST CITATION NUMBER IN LSEL.	OSSF133
C END	OSSF134

PROGRAM SEARCH(INPUT,OUTPUT,TAPE1,TAPE2,	SEARCH 2
, TAPE51=INPUT,TAPE51=OUTPUT)	SEARCH 3
C SEARCH PROGRAM: PART OF THE FAMULUS SYSTEM.	SEARCH 4
COMMON /CORE/NCHW,BLANKS,PLANK	SEARCH 5
INTEGER BLANKS,PLANK	SEARCH 6
COMMON /FAM/SLASH,STAR,COMMA,ZERO,MXCHAR,MXCODES,MXWDES,MXCHW,	SEARCH 7
, MXFREE,MXWFRE	SEARCH 8
INTEGER SLASH,STAR,COMMA,ZERO	SEARCH 9
COMMON /FIEL/FIELD(10),IFIELD(10),NFIELD,KDESCR	SEARCH10
INTEGERCK FIELD	SEARCH11
COMMON /IPT/CARD(10),KOLUM,HOLD,ENDCAR,BLACAR,CONCAR,ERRIN,NWC,	SEARCH12
, MXFL,MXCAR,NCARIN,KHR,KIN,LIT,LITX,SYM(5),NUM,NCSYM,CATALOG,	SEARCH13
, KARA,KARB	SEARCH14
INTEGER CARD,SYM	SEARCH15
LOGICAL HOLD,ENDCAR,BLACAR,CONCAR,ERRIN,CATALOG	SEARCH16
COMMON /TEST/ITEST	SEARCH17
COMMON TEXT(250),LTEXT(50),NCTEXT(50),NTEXT,	SEARCH18
, JTERM(60),MODFLD(60),LOCATE(60),NTERM,MAYBE(64),KODET(64),	SEARCH19
, KUKU(150),NKUKU,LUKE(150),MLUKU,LIST(150),NLIST,	SEARCH20
, KSERCH(10),NSERCH,KFLD(10),MXTEXT,MXCTXT,MXTERM,MXLUKE	SEARCH21
INTEGER IFA(13),IL(13),CIT(3,180)	SEARCH22
INTEGER JLAB(10),LSFL(100)	07/18/76
LOGICAL WHIT,SHORT,FCHOI,NONO,MEMBER	07/18/77
DATA MXSEL / 100 /	07/18/78
DATA MXWID,MLWID / 128,20 /	07/18/79
DATA LCITA,LCITB,NWRIT/3*0/,WRIT/,FALSE, /	SEARCH25
C INITIALIZATION,	SEARCH26
NWIDTH = 100	
CALL USHER(IDA,KERR)	SEARCH27
KE=1	SEARCH28
IF(KERR,NE,0) GO TO 900	SEARCH24
NTEXT=0	SEARCH30
NTERM=0	SEARCH31
NLIST=0	SEARCH32
NSERCH=0	SEARCH33
NTIMES=0	SEARCH34
NKUKU=0	SEARCH35
NLUKE=0	SEARCH36
NONO = .FALSE.	07/18/80
FCHOI = .FALSE.	07/18/81
DO 5 I=1,10	SEARCH37
KSERCH(I)=0	SEARCH38
3 KFLD(I)=0	SEARCH39
MTA=1	SEARCH40
MTB=2	SEARCH41
CALL REWARD(MTA)	SEARCH42
NTERM=60	SEARCH43
MXTEXT=50	SEARCH44
MXLUKE=150	SEARCH45
MXCTXT=250*NCHW	SEARCH40
SHORT=.FALSE.	SEARCH47
C SET KODET TO INDICATE ALPHABETIC, NUMERIC, AND MINUS CHARACTERS.	SEARCH48
DO 8 J=1,64	SEARCH49
KODET(J)=0	SEARCH50
IJ=IDEAL(J-1)	SEARCH51
IF(IJ.EQ.2,OR,IJ.EQ.3,OR,IJ.EQ.8) KODET(J)=1	SEARCH52
C THE NEXT STATEMENT PUTS THE APOSTROPHE IN THE SET OF DESCRIPTOR CHARS.	SEARCH53

IF (J-1.EQ.KHAR(1H",1)) KODET(J)=1	SEARCH54
C---ADD SEMICOLON AS DELIMITER IN SET OF DESCRIPTOR CODE CHARACTERS	07/18/82
IF (J-1.EQ.KHAR(1H",1)) KODET(J) = 1	07/18/83
8 CONTINUE	SEARCH55
CALL KORAN(5HKODET,KODET,64)	SEARCH56
C PREFACE RECORD,	SEARCH57
CALL IOID(0,MTA,ID,FIELD,IFIELD,NFIELD,KDESCR)	SEARCH58
CALL IDPRIN(ID,FIELD,IFIELD,NFIELD,KDESCR,10,10HINPUT TAPE)	SEARCH59
CALL SIMILE(1,ID,0,MXFREE,IDA,0,MXFREE,LIKE)	SEARCH60
KE=3	SEARCH61
IF (LIKE.NE.1) GO TO 900	SEARCH62
CALL REWAND(MTA)	SEARCH63
CALL PAGER(-5)	SEARCH64
C BEGIN MAJCH SEARCH LOOP.	SEARCH65
20 CONTINUE	SEARCH66
NTIMES=NTIMES+1	SEARCH67
WRIT=.FALSE.	SEARCH68
IF (ITEST.NE.0) CALL PAGER(0)	SEARCH69
C NEXT STATEMENT.	SEARCH70
30 KE=9	SEARCH71
IF (ERRIN.AND.NTIMES.LE.1) GO TO 900	SEARCH72
IF (ITEST.NE.0) CALL PAGER(-4)	SEARCH73
CALL SETSYM	SEARCH74
IF (ENDCAR) GO TO 200	SEARCH75
CALL FABRIC(KON)	SEARCH76
IF (KOLUM.NE.1) CALL DIVEST	SEARCH77
IF (KON.NE.2) GO TO 50	SEARCH78
C /FIELDS/ STATEMENT.	SEARCH79
KE=8	SEARCH80
IF (NTIMES.NE.1) GO TO 900	SEARCH81
CALL GETLAB(KFLD,1,NFIELD,NSERCH,KERR)	SEARCH82
KE=4	SEARCH83
IF (KERR.GT.0) GO TO 900	SEARCH84
DO 40 K=1,NSERCH	SEARCH85
L=LOOK(FIELD,NFIELD,KFLD(K))	SEARCH86
KE=5	SEARCH87
IF (L.GT.NFIELD) GO TO 900	SEARCH88
40 KSERCH(L)=1	SEARCH89
CALL KORAN(6HKSERCH,KSERCH,NFIELD)	SEARCH90
GO TO 30	SEARCH91
50 IF (KON.NE.14) GO TO 70	SEARCH92
C /WRITE TAPE/ STATEMENT.	SEARCH93
IF (NWRIT.EQ.0) WRIT=.TRUE.	SEARCH94
NWRIT=NWRIT+1	SEARCH95
IF (.NOT.WRIT) GO TO 60	SEARCH96
CALL REWAND(MTB)	SEARCH97
CALL IOID(1,MTB,ID,FIELD,IFIELD,NFIELD,KDESCR)	SEARCH98
IF (ITEST.EQ.0) GO TO 30	SEARCH99
CALL IDPRIN(ID,FIELD,IFIELD,NFIELD,KDESCR,11,11HOUTPUT TAPE)	SEARCH100
GO TO 30	SEARCH101
60 CALL PAGER(3)	SEARCH102
PRINT 1000	SEARCH103
1000 FORMAT(/65H THE RESULTS OF ONLY ONE SEARCH MAY BE WRITTEN ON THE	SEARCH104
OUTPUT TAPE, 63(14*)/1X)	SEARCH105
GO TO 30	SEARCH106
70 IF (KON.NE.18) GO TO 65	07/18/84
C /NUMBERS/ STATEMENT	SEARCH108

SHORT=.TRUE.	SEARC109
GO TO 30	SEARC110
65 IF (KON,NE. 5) GO TO 85	07/18/85
C /SELECT/	07/18/86
CALL GETNUM(LSEL,1,MXSEL,N,KERR)	07/18/87
KE = 3	07/18/88
IF (KERR.GT. 0) GO TO 900	07/18/89
LATE = IARS(LSEL(1))	07/18/90
CALL KORAN(4HSEL,LSEL,N+1)	07/18/91
GO TO 30	07/18/92
85 IF (KON,NE. 20) GO TO 90	07/18/93
C-----THE MODIFICATIONS LABELLED 07/10/73 WERE ADDED BY RJ OLSON-----	07/18/94
C THESE CHANGES REFLECT ADDING STANDARD WIDTH OPTION (AS IN GALLEY)	07/18/95
C A#2 FIELD CHOICE CAR AS SUPPLIED BY JIM CLARKE OF BERKLEY	07/18/96
C CHANGES ALSO MADE TO PRETTY AND FABRIC AT THIS TIME,	07/18/97
C /FIELD CHOICE/	07/18/98
FCHOI = .TRUE.	07/18/99
CALL GETLAB(JLAB,1,NFIELD,NCHOI,KERR)	07/18100
IF (KERR.GT. 0) GO TO 900	07/18101
DO 105 K = 1,NCHOI	07/18102
L = LOOK(FIELD,NFIELD,JLAB(K))	07/18103
IF (L.GT. NFIELD) GO TO 900	07/18104
JLAB(K) = L	07/18105
105 CONTINUE	07/18106
GO TO 30	07/18107
90 IF (KON,NE. 15) GO TO 95	07/18108
C /WIDTH/	07/18109
CALL GETNUM(LIST,1,1,N,KERR)	07/18110
KE = 4	07/18111
IF (KERR.GT. 0) GO TO 900	07/18112
MWIDTH = MAX0(MIN0(LIST(1),MXWID),MNWID)	07/18113
GO TO 30	07/18114
95 IF (KON,NE. 21) GO TO 75	07/18115
C /NO NUMBERS/ OPTION TO DELETE NUMBERS ON OUTPUT FROM PRETTY	07/18116
NONO = .TRUE.	07/18117
GO TO 30	07/18118
75 KE=6	SEARC111
IF (KON,NE.10) GO TO 900	SEARC112
C /SEARCH/ STATEMENT, CONSTRUCT POLISH STRING FROM FORMULA,	SEARC113
IF (NSERCH.GT.0) GO TO 80	SEARC114
KE=7	SEARC115
IF (KDESCR.LE.0) GO TO 900	SEARC116
KSERCH(KDESCR)=1	SEARC117
NSERCH=1	SEARC118
CALL KORAN(6HKSERCH,KSERCH,NFIELD)	SEARC119
80 LCITA=0	SE4hC120
LCITB=0	SEARC121
C--START NEW PAGE FOR EACH SEARCH	07/18119
CALL PAGER(-50)	07/18120
READ(MTA)	SEARC122
CALL POLISH	SEARC123
IF (ERRIN) GO TO 800	SEARC124
C BEGIN THE SEARCH.	SEARC125
C READ NEXT CITATION, SEARCH IT, DETERMINE IF IT IS WANTED.	SEARC126
100 IF (ITEST, E, 0) CALL PAGER(-4)	SEARC127
CALL KORAN(5HLCITA,LCITA+1,1)	SEARC128
,ALL IOTAPE(),MTA,CIT,LAST)	SEARC129

IF(LAST,FC,1) GO TO 150	SEARC130
LCITA=LCITA+1	SEARC131
IF (.NOT. MEMBER(LSEL,MXSEL,LCITA)) GO TO 100	07/18121
CALL MOVE(0,LOCATE,-NTERM)	SEARC132
C SEARCH ONE FIELD AT A TIME.	SEARC133
EQ 123 K=1,NFIELD	SEARC134
CALL KORAN(5HFIELD,K,1)	SEARC135
NC=CIT(1,K)	SEARC136
IF(NC,LE,0,OR,KFLD(K),LE,0,AND,KSERCH(K),LE,0) GO TO 120	SEARC137
CALL NIMROD(K,CIT)	STARC138
CALL KORAN(6HLOCATE,LOCATE,INTERM)	STARC139
120 CONTINUE	STARC140
C EVALUATE THE BOOLEAN FORMULA (IN LUKE) USING LOCATE.	SEARC141
CALL BOOLE(IFY)	STARC142
IF(IFY,NE,1) GO TO 130	SEARC143
LCITB=LCITB+1	SEARC144
IF(SHORT) GO TO 124	SEARC145
JCIT = LCITA	07/18122
IF (NONO) JCIT = 0	07/18123
CALL PRETTY(CIT,JCIT,0,NWIDTH,FCHOI,ULAB,VCHOI)	07/18124
GO TO 126	SEARC147
124 CALL PAGER(1)	STARC148
PRINT 5000, LCITA	SEARC149
5000 FORMAT (16H CITATION NUMBER,16)	SEARC150
120 CONTINUE	SEARC151
IF(.NOT.WRIT) GO TO 130	SEARC152
CALL IOTAPE(1,MTB,CIT,0)	SEARC153
130 CONTINUE	STARC154
GO TO 100	SEARC155
C END OF SEARCH.	STARC156
150 CALL REWARD(NTA)	SEARC157
C---WRITE SEARCH STATISTICS ON NEW PAGE	07/18125
CALL PAGER(0)	07/18126
CALL PAGER(2)	SEARC159
PRINT 2000, LCITB	SEARC160
2000 FORMAT(1X,14,38H CITATIONS SATISFY THE SEARCH REQUEST,/1X)	SEARC161
IF(.NOT.WRIT) GO TO 160	SEARC162
CALL IOTAPE(1,MTB,CIT,1)	SEARC163
CALL REWARD(MTB)	SEARC164
CALL IDPRIN(ID,FIELD,IFIELD,NFIELD,KDESCR,11,11HOUTPUT TAPE)	SEARC165
CALL PAGER(2)	SEARC166
PRINT 3000,LCITB	SEARC167
3000 FORMAT(10H THERE ARE,15,30H CITATIONS ON THE OUTPUT TAPE,/1X)	STARC168
160 CONTINUE	SEARC169
GO TO 20	SEARC170
C NORMAL END.	SEARC171
200 CALL PAGER(0)	SEARC172
CALL PAGER(1)	SEARC173
PRINT 4000	SEARC174
4000 FORMAT(18H END OF SEARCHING,)	SEAYC175
CALL PAGER(11)	SEARC176
PRINT 8999	SEARC177
8999 FORMAT(/	SEARC178
1 /3(20X 17H XXXXX X X)	STARC179
2 /3(20X 17HX X X X)	SEARC180
3 /3(20X 17HX X X X)	SEARC181
4 /3(20X 17HX X X)	STARC182

5	/3(20< 17HX	X	X	X	)	SEARC183
6	/3(20< 17HX	X	X	X	)	SEARC184
7	/3(20< 17H XXXXX	X			4)	SEARC185
8	//)					SEARC186
	STOP					SEARC187
C	ERROR IN /SEARCH/ STATEMENT.					SEARC188
800	IF (ENDCAR) GO TO 200					SEARC189
	ERRIN=.FALSE.					SEARC190
	IF (CONCAR) GO TO 20					SEARC191
	CALL INCARD					SEARC192
	CALL DIVERST					SEARC193
	GO TO 800					SEARC194
C	FATAL ERRORS.					SEARC195
900	IF (.NOT. CONCAR) CALL DIVERST					SEARC196
	CALL PAGE(2)					SEARC197
	GO TO (901,902,903,904,905,					SEARC198
	906,907,908,909),KE					SEARC199
901	PRINT 9001					SEARC200
9001	FORMAT(10H NO INPUT., 118(1H*))					SEARC201
	GO TO 950					SEARC202
902	PRINT 9002					SEARC203
9002	FORMAT(34H THE FIRST STATEMENT MUST BE /ID/, 94(1H*))					SEARC204
	GO TO 950					SEARC205
903	PRINT 9003					SEARC206
9003	FORMAT(29H TAPE II DOES NOT MATCH /ID/, 99(1H*))					SEARC207
	GO TO 950					SEARC208
904	PRINT 9004					SEARC209
9004	FORMAT(19H ERROR IN /FIELDS/, 109(1H*))					SEARC210
	GO TO 950					SEARC211
905	PRINT 9005					SEARC212
9005	FORMAT(30H MISPELLED LABEL IN /FIELDS/, 98(1H*))					SEARC213
	GO TO 950					SEARC214
906	PRINT 9006					SEARC215
9006	FORMAT(22H EXTRANEOUS STATEMENT., 106(1H*))					SEARC216
	GO TO 950					SEARC217
907	PRINT 9007					SEARC218
9007	FORMAT(32H A /FIELDS/ STATEMENT IS NEEDED, 96(1H*))					SEARC219
	GO TO 950					SEARC220
908	PRINT 9008					SEARC221
9008	FORMAT(33H MORE THAN ONE /FIELDS/ STATEMENT, 95(1H*))					SEARC222
	GO TO 950					SEARC223
909	PRINT 9009					SEARC224
3009	FORMAT(13H INPUT ERROR , 115(1H*))					SEARC225
950	PRINT 9050, KE					SEARC226
9050	FORMAT(118X, 6HSEARCH 14)					SEARC227
	STOP					SEARC228
C	CIT(450)	CONTENTS OF THE CITATION CURRENTLY BEING PROCESSED.				SEARC229
C	ID(13)	IDENTIFICATION FROM INPUT TAPE.				SEARC230
C	IIA(13)	IDENTIFICATION IN /ID/.				SEARC231
C	JTERM(60)	JTERM(J) POINTS TO THE POSITION IN TEXT OF THE				SEARC232
C		CHARACTER STRING COMPRISING DESCRIPTOR NUMBER J.				SEARC233
C	KFLD(10)	KFLD(K)=1 IF K OCCURS AT LEAST ONCE IN MODFL.				SEARC234
C		KFLD(K)=0 IF K DOES NOT OCCUR IN MODFL.				SEARC235
C	KODET(64)	INDICATES WHICH OF THE 64 POSSIBLE CHARACTERS ARE				SEARC236
C		ALPHABETIC (A-Z), NUMERIC (0-9), OR HYPHEN (-), I.E.,				SEARC237
C		IF THE CHARACTER CODE K REPRESENTS ONE OF THESE 37				SEARC238
C		CHARACTERS, THEN KODET(K+1)=1. IF NOT, KODET(K+1)=0.				SEARC239

C		THIS ARRAY IS THUS CONSTANT FOR A GIVEN COMPUTER.	SEARC240
C	KSERCH(10)	KSERCH(J)=1 IF FIELD J IS IN /FIELDS/. KSERCH(J)=0 IF NOT.	SEARC241
C			SEARC242
C	KUKE(150)	OPERATOR-OPERAND LIST FROM /SEARCH/, I.E., KUKE	SEARC243
C		CONTAINS THE ESSENCE OF THE FORMULA, SAME CODE AS LUKE,	SEARC244
C		PLUS LEFT PARENTHESIS=-4, RIGHT PARENTHESIS=-5.	SEARC245
C	LCITA	NUMBER OF THE CURRENT CITATION FROM THE INPUT TAPE.	SEARC246
C	LCITB	NUMBER OF CITATIONS THAT HAVE SATISFIED THE SEARCH REQUEST.	SEARC247
C			SEARC248
C	LIST(150)	PUSH-DOWN STACK, USED TO CONVERT KUKE TO LUKE, AND TO EVALUATE (VIA LUKE) THE FORMULA FOR A CITATION.	SEARC249
C			SEARC250
C	LOCATE(60)	INDICATES RESULTS OF SEARCHING A CITATION.	SEARC251
C		LOCATE(J)=1 IF J-TH TERM WAS FOUND IN THE CITATION,	SEARC252
C		LOCATE(J)=0 IF NOT.	SEARC253
C	LTEXT(50)	LTEXT(J) POINTS TO THE FIRST CHARACTER OF THE J-TH SUBSTRING IN TEXT.	SEARC254
C			SEARC255
C	LUKE(150)	POLISH STRING REPRESENTING THE FORMULA (EXPRESSION) IN /SEARCH/. OPERANDS ARE REPRESENTED BY POSITIVE INTEGERS BETWEEN 1 AND NTERM, OPERATORS ARE REPRESENTED AS FOLLOWS: OR=-1, AND=-2, NOT=-3, END OF FORMULA=0.	SEARC256
C			SEARC257
C			SEARC258
C			SEARC259
C			SEARC260
C	MAYBE(64)	INDICATES, DURING PROCESSING OF A FIELD, THE SET OF ALL CHARACTERS WHICH OCCUR IN THE FIRST CHARACTER POSITION OF TERMS BEING SEARCHED FOR IN THAT FIELD.	SEARC261
C		I.E., IF MAYBE(K+1)=1, THEN THE CHARACTER WHOSE CODE IS K IS THE LEAD CHARACTER OF SOME SUBSTRING OF TEXT BEING SEARCHED FOR, OTHERWISE, MAYBE(K+1)=0.	SEARC262
C			SEARC263
C			SEARC264
C			SEARC265
C	MODFLD(60)	MODFLD(J)=0 IF J-TH TERM HAS NO FIELD MODIFIER.	SEARC266
C		MODFLD(J)=K IF FIELD K MODIFIES THE J-TH TERM.	SEARC267
C			SEARC268
C	MTA	INPUT TAPE.	SEARC269
C	MTB	OUTPUT TAPE.	SEARC270
C	MXCTXT	MAX. TOTAL NUMBER OF CHARACTERS IN TEXT ARRAY, (2000)	SEARC271
C	MXLUKE	MAXIMUM NUMBER OF SYMBOLS IN A FORMULA, (150)	SEARC272
C	MXTERM	MAXIMUM NUMBER OF UNIQUE TERMS IN A FORMULA, (60)	SEARC273
C		A TERM CONSISTS OF A POINTER TO TEXT (JTERM) TOGETHER WITH A FIELD MODIFIER (MODFLD) WHICH MAY BE ZERO.	SEARC274
C			SEARC275
C	MYTEXT	MAX. NUMBER OF UNIQUE SUBSTRINGS IN TEXT ARRAY, (50)	SEARC276
C	NCTEXT(50)	NCTEXT(J) IS THE NUMBER OF CHARACTERS IN J-TH SUBSTRING IN TEXT ARRAY.	SEARC277
C			SEARC278
C	NKUKE	NUMBER OF ENTRIES IN KUKE.	SEARC279
C	NLIST	LENGTH OF LIST ARRAY, ALSO POINTS TO CURRENT STACK POSITION.	SEARC280
C			SEARC281
C	NLUKE	NUMBER OF ENTRIES IN LUKE.	SEARC282
C	NSERCH	NUMBER OF FIELD LABELS IN /FIELDS/, (NUMBER OF ONES IN KSERCH.)	SEARC283
C			SEARC284
C	NTERM	NUMBER OF TERMS (LENGTH OF JTERM AND OF MODFLD)	SEARC285
C	NTIMES	NUMBER OF SEARCH REQUESTS.	SEARC286
C	NTEXT	NUMBER OF SUBSTRINGS IN TEXT ARRAY.	SEARC287
C	NWRIT	NUMBER OF TIMES /WRITE TAPE/ HAS BEEN ENCOUNTERED SO FAR.	SEARC288
C			SEARC289
C	TEXT(250)	CHARACTER STRING CONTAINING AS SUBSTRINGS ALL TERMS IN /SEARCH/. DUPLICATES ARE ELIMINATED.	SEARC290
C			SEARC291
C	WRIT	TRUE IFF /WRITE TAPE/ IS PRESENT IN INPUT.	SEARC292
C	END		SEARC293

PROGRAM SORT(INPUT,OUTPUT,TAPE1,TAPE2,TAPE3,TAPE4,TAPE5,TAPE6,	SORT	2
, TAPE50=INPUT,TAPE51=OUTPUT)	SORT	3
C SORT PROGRAM, PART OF THE FAMULUS SYSTEM.	SORT	4
COMMON /CORE/NCHW,BLANKS,BLANK	SORT	5
INTEGER BLANKS,BLANK	SORT	6
COMMON /FAM/SLASH,STAR,COMMA,ZERO,MXCHAR,MXCLUDES,MXWLES,MXCHW,	SORT	7
, MXFRE,MXWFRE	SORT	8
INTEGER SLASH,STAR,COMMA,ZERO	SORT	9
COMMON /FIEL/FIELD(10),IFIELD(10),NFIELD,MATCH	SORT	10
INTEGER FIELD	SORT	11
COMMON /INPT/CAR(10),KOLIM,HOLD,ENDCAR,BLACAR,CONCAR,ERRIN,MXC,	SORT	12
MXFL,MXCAR,NCARIN,KHR,KIN,LIT,LITX,SYM(5),NONE,MCSYM,CATLOG,	SORT	13
KARATKAR	SORT	14
INTEGER KAR,SYM	SORT	15
LOGICAL HOLD,ENDCAR,BLACAR,CONCAR,ERRIN,CATLOG	SORT	16
COMMON /PAGING/IPAGE,LINES,MAXLIN,PAGHED(13)	SORT	17
INTEGER PAGHED	SORT	18
COMMON /TEST/ITEST	SORT	19
C	SORT	20
COMMON LCIT(300),CIT(10000)	SORT	21
INTEGER CIT	SORT	22
INTEGER CITM(3,180,2),CITIN(3,180)	SORT	23
INTEGER KANK(300),NSTRN(2,2),MTAPE(2,2),JUNCIT(2,2),LASTC(2)	SORT	24
INTEGER IIA(13),I(13),JFIELD(10),FLD(10),LSEL(100)	SORT	25
LOGICAL ENDIN,FINAL,MEMBER	SORT	26
DATA MXSEL/100/,MXCIT/300/,MXWC/540/	SORT	27
DATA LENCIT/10000/	SORT	28
C INITIALIZATION	SORT	29
CALL USHER(I)A,KERR)	SORT	30
NE=1	SORT	31
IF(KERR,NE,0) GO TO 903	SORT	32
ITST=ITEST	SORT	33
ITEST=0	SORT	34
IF(ITST,0,1) ITEST=ITST	SORT	35
C SPECIAL SETTINGS OF THE DEBUG SWITCH ITEST...	SORT	36
C ITEST=1, START DEBUG PRINTING AT VERY BEGINNING.	SORT	37
C ITEST=2, START DEBUG PRINTING AT END OF INITIALIZATION.	SORT	38
C ITEST=3, START DEBUG PRINTING AT BEGINNING OF MERGE PHASE.	SORT	39
C ITEST=IPASS+300, DO DEBUG PRINTING ONLY DURING MERGE PASS NO. IPASS.	SORT	40
NFLD=0	SORT	41
LCITOU=0	SORT	42
LCITIN=0	SORT	43
LCITOU=0	SORT	44
IPASS=0	SORT	45
COMPR=0	SORT	46
FINAL=.FALSE.	SORT	47
ENDIN=.FALSE.	SORT	48
MTIN=1	SORT	49
MTOUT=2	SORT	50
MTAPE(1,1)=3	SORT	51
MTAPE(2,1)=4	SORT	52
MTAPE(1,2)=5	SORT	53
MTAPE(2,2)=6	SORT	54
CALL REWARD(MTIN)	SORT	55
CALL REWARD(MTOUT)	SORT	56
DO 10 J=1,2	SORT	57
DO 10 I=1,2	SORT	58

CALL REWIND(MTAPE(I,J))	SORT	59
NUMCIT(I,J)=0	SORT	60
10 NSTRING(I,J)=0	SORT	61
LATE=100000	SORT	62
LSEL(1)=1	SORT	63
LSEL(2)=-LATE	SORT	64
LSEL(3)=1	SORT	65
C NEXT STATEMENT,	SORT	66
2, ,ALL SETSYM	SORT	67
IF(ENLCAL) GO TO 50	SORT	68
CALL FA3RIC(KON)	SORT	69
KE=2	SORT	70
IF(ERRIN) GO TO 900	SORT	71
IF(KON.NF.2) GO TO 30	SORT	72
C FIELDS STATEMENT,	SORT	73
CALL GETLAB(FLD,1,10,NFLD,KERR)	SORT	74
KE=3	SORT	75
IF(KERR.GT.0) GO TO 900	SORT	76
CALL TALMUD(3HFLD,FLD,0,NFLD*NCHW)	SORT	77
GO TO 20	SORT	78
C SELECT STATEMENT,	SORT	79
30 IF(KON.NF.3) GO TO 40	SORT	80
CALL GETNUM(LSEL,1,MXSEL,N,KERR)	SORT	81
LATE=IABS(LSEL(N))	SORT	82
KE=4	SORT	83
IF(KERR.GT.0) GO TO 930	SORT	84
CALL KORAN(4HLSEL,LSEL,N)	SORT	85
GO TO 20	SORT	86
40 KE=5	SORT	87
GO TO 900	SORT	88
C FINAL INITIALIZATION,	SORT	89
50 KE=6	SORT	90
IF(NFLD.LE.0) GO TO 900	SORT	91
IF(ITST.EQ.2) ITST=ITST	SORT	92
CALL IOI(0,MTIN,ID,FIELD,IFIELD,NFIELD,KDESCR)	SORT	93
CALL IDPRIN(ID,FIELD,IFIELD,NFIELD,KDESCR,10,10HINPUT TAPE)	SORT	94
CALL SIMILE(1,ID,0,MXFREE,IDA,0,MXFREE,LIKE)	SORT	95
KE=7	SORT	96
IF(LIKE.NE.1) GO TO 900	SORT	97
KE=8	SORT	98
IF(NFLD.GT.NFIELD) GO TO 900	SOHT	99
CALL MOVE(0,JFIELD,-NFIELD)	SORT	100
DO 60 K=1,NFLD	SORT	101
L=LOOK(FIELD,NFIELD,FLD(K))	SOHT	102
KE=9	SORT	103
IF(L.GT.NFIELD) GO TO 900	SORT	104
IF(LOOK(JFIELD,K-1,L).LT.K) GO TO 900	SORT	105
60 JFIELD(K)=L	SORT	106
M=NFLD	SORT	107
DO 65 K=1,NFIELD	SORT	108
L=LOOK(JFIELD,NFLD,K)	SORT	109
IF(L.LE.NFLD) GO TO 65	SORT	110
M=M+1	SORT	111
JFIELD(M)=K	SORT	112
65 CONTINUE	SOHT	113
CALL KORAN(6HJFIELD,JFIELD,NFIELD)	SOHT	114
DO 70 K=1,NFIELD	SORT	115

KF=IFIELD(K)	116
IFIELD(K)=JFIELD(K)	117
70 JFIELD(K)=KF	118
CALL IOTI(1,MTOUT,ID,FIELD,IFIELD,NFIELD,KDESCR)	119
CALL IDPRIN(ID,FIELD,IFIELD,NFIELD,KDESCR,11,11HOUTPUT TAPE)	120
CALL PAGER(-3)	121
C SORT PHASE,	122
MTA=2	123
MTB=1	124
KT=2	125
LCI=0	126
C NEXT STRING,	127
100 NCIT=0	126
LC=1	129
KT=3-KT	130
C NEXT CITATION,	131
110 CALL IOTAPE(0,MTIN,CITIN,LAST)	132
ENDIN=LAST.EQ.1 OR LCITIN GT LAST	133
IF(ENDIN) GO TO 120	134
LCITIN=LCITIN+1	135
IF(.NOT.MEMBER(LSEL,MXSEL,LCITIN)) GO TO 110	136
NCIT=NCIT+1	137
LCIT(NCIT)=LC	138
CALL COPCIT(CITIN,CIT(LC),NC)	139
LC=LC+NC	140
LCITOU=LCITOU+1	141
IF(NCIT.LT.MXCIT.AND.LC+MXWC.LE.LENCIT) GO TO 110	142
C SORT CIT,	143
120 IF(NCIT.LE.0) GO TO 140	144
CALL SHELLS(CIT,KRANK,NCIT)	145
C-----	JUT
C-----CALL SHELL CHANGE HERE TO SHELLS BY JOHN R. TADLOCK NOV. 1975,	JRT
NSTRNG(KT,MTB)=NSTRNG(KT,MTB)+1	146
NUMCIT(KT,MTB)=NUMCIT(KT,MTB)+NCIT	147
IF(ENDIN.AND.NSTRNG(1,MTB)+NSTRNG(2,MTB).EQ.1) FINAL=.TRUE.	148
IF(FINAL) MTAPE(KT,MTB)=MTOUT	149
C WRITE SORTED CITATIONS ONTO TAPE,	150
DO 130 J=1,NCIT	151
LCI=LCI+1	152
K=KRANK(J)	153
L=LCIT(K)	154
CALL IOTAPE(1,MTAPE(KT,MTR),CIT(L),0)	155
IF(ITEST,NE,0) CALL PRNCIT(CIT(L),LCI,0,128)	156
130 CONTINUE	157
CALL IOTAPE(1,MTAPE(KT,MTB),0,1)	158
CALL PAGER(1)	159
PRINT 3000, NCIT,NSTRNG(KT,MTB),MTAPE(KT,MTB),LCITOU	160
3000 FORMAT(1X,I4,20H CITATIONS IN STRING,13,6H ON UNIT,12,1H,,	161
16,17H CITATIONS SO FAR)	162
IF(.NOT.ENDIN) GO TO 100	163
C END OF INTERNAL SORT. ALL INPUT TAPE CITATIONS ARE PROCESSED,	164
140 CONTINUE	165
CALL REWAND(MTIN)	166
DO 150 J=1,2	167
MT=MTAPE(J,MTB)	168
CALL IOTAPE(1,MT,0,1)	169
150 CALL REWAND(MT)	170

NCITOU=LCITOU	SORT 171
CALL PAGER(4)	SORT 172
PRINT 4000,(MTAPE(J,MTB),NUMCIT(J,MTB),NSTRNG(J,MTB), J=1,2)	SORT 173
4000 FORMAT(/24H END OF INTERNAL SORTING,	SORT 174
, 2(/5H UNIT 12,16,10H CITATIONS,14,8H STRINGS))	SORT 175
IF(FINAL) SO TO 500	SORT 176
C MERGE PHASE.	SORT 177
C INITIALIZE.	SORT 178
IF(ITST.EQ,3) ITEST=ITST	SORT 179
FINAL=,FALSE,	SORT 180
LCITOU=0	SORT 181
NCOMPR=0	SORT 182
IPASS=0	SORT 183
C BEGINNING OF PASS.	SORT 184
200 IPASS=IPASS+1	SORT 185
IF(ITST.NE,3) ITEST=0	SORT 186
IF(ITST.EQ,IPASS+300) ITEST=ITST	SORT 187
MTA=3-MTA	SORT 188
MTB=3-MTB	SORT 189
IF(NSTRNG(1,MTA).GT.1.OR,NSTRNG(2,MTA).GT.1) GO TO 210	SORT 190
MTAPE(1,MTB)=MTOUT	SORT 191
FINAL=,TRUE,	SORT 192
210 IF(ITEST.NE,0) CALL PAGER(0)	SORT 193
CALL SPOT(IPASS,MTA,MTB,FINAL,4HIIIL,24HIPASS MTA MTB FINAL)	SORT 194
DO 220 K=1,2	SORT 195
NSTRNG(K,MTB) = 0	SORT 196
NUMCIT(K,MTB)=0	SORT 197
DO 220 J=1,2	SORT 198
IF(MTAPE(J,K).NE,MTOUT) CALL REWAND(MTAPE(J,K))	SORT 199
220 CONTINUE	SORT 200
CALL PAGER(-4)	SORT 201
CALL PAGER(1)	SORT 202
PRINT 2000, IPASS,MTAPE(1,MTA),MTAPE(2,MTA),	SORT 203
, MTAPE(1,MTB),MTAPE(2,MTB)	SORT 204
2000 FORMAT(5H PASS 13,5X,6HINPUTS 212,5X,7HOUTPUTS 212)	SORT 205
ISTRNG=0	SORT 206
KT=2	SORT 207
C BEGINNING OF STRING.	SORT 208
230 KT=3-KT	SORT 209
ISTRNG=ISTRNG+1	SORT 210
LASTC(1)=-1	SORT 211
LASTC(2)=-1	SORT 212
LENSTR=0	SORT 213
CALL SPOT(ISTRNG,KT,0,0,4HIIIII,24HISTRNGKT)	SORT 214
C COMPARISON OF A PAIR OF CITATIONS.	SORT 215
240 NEXT=0	SORT 216
DO 250 K=1,2	SORT 217
IF(LASTC(K).NE,-1) GO TO 245	SORT 218
CALL IOTAPE(0,MTAPE(K,MTA),CITM(1,1,K),LASTC(K))	SORT 219
IF(LASTC(K).EQ,0) NUMCIT(K,MTA)=NUMCIT(K,MTA)-1	SORT 220
245 IF(LASTC(K).EQ,1) NEXT=3-K	SORT 221
250 CONTINUE	SORT 222
IF(LASTC(1).EQ,1.AND,LASTC(2).EQ,1) GO TO 280	SORT 223
IF(NEXT.NE,0) GO TO 260	SORT 224
LEAD=LEADER(CITM(1,1,1),CITM(1,1,2))	SORT 225
NCOMPR=NCOMPR+1	SORT 226
NEXT=2	SORT 227

IF(LEAD.EQ.-1) NEXT=1	SORT 228
260 IF(ITEST.EQ.0) GO TO 265	SORT 229
CALL SPOT(NEXT,LEAD,NCOMPR,LENSTR,4H1111,	SORT 230
24HNEXT LEAD NCMPLLENSTR)	SORT 231
CALL KORAN(6HNUMCIT,NUMCIT,4)	SORT 232
CALL KORAN(5HLASTC,LASTC,2)	SORT 233
265 CALL IOTAPE(1,MTAPE(KT,MTR),CITM(1,1,NEXT),0)	SORT 234
LASTC(NEXT)=-1	SORT 235
LENSTR=LENSTR+1	SORT 236
IF(.NOT.FINAL) GO TO 270	SORT 237
LCITOU=LCITOU+1	SORT 238
IF(ITEST.EQ.0) GO TO 270	SORT 239
CALL PRNCIT(CITM(1,1,NEXT),LCITOU,0,128)	SORT 240
270 CONTINUE	SORT 241
GO TO 240	SORT 242
C END OF STYLING.	SORT 243
280 NSTRNG(KT,MTB)=NSTRNG(KT,MTB)+1	SORT 244
NUMCIT(KT,MTR)=NUMCIT(KT,MTB)+LENSTR	SORT 245
CALL IOTAPE(1,MTAPE(KT,MTR),0,1)	SORT 246
CALL KORAN(6HNSTRNG,NSTRNG,4)	SORT 247
CALL KORAN(6HNUMCIT,NUMCIT,4)	SORT 248
CALL PAGER(2)	SORT 249
PRINT 5000, ISTRNG,LENSTR,NCOMPR,MTAPE(KT,MTB),NUMCIT(KT,MTB),	SORT 250
NSTRNG(KT,MTB)	SORT 251
5000 FORMAT(7H STRING 13,16,10H CITATIONS,17,19H COMPARISONS SO FAR,	SORT 252
10X,4HTAPE 12,4H HAS 16,13H CITATIONS IN 13,8H STRINGS)	SORT 253
IF(NUMCIT(1,MTA)+NUMCIT(2,MTA).GT.0) GO TO 230	SORT 254
C END OF PASS.	SORT 255
DO 290 K=1,2	SORT 256
290 CALL IOTAPE(1,MTAPE(K,MTR),0,1)	SORT 257
IF(.NOT.FINAL) GO TO 200	SORT 258
C END OF MERGE.	SORT 259
500 CALL REWABD(MTOUT)	SORT 260
CALL PAGER(-3)	SORT 261
CALL PAGER(2)	SORT 262
PRINT 8000, LCITOU	SORT 263
8000 FORMAT(16H END OF PROGRAM,712H HAVE SORTED 16,10H CITATIONS)	SORT 264
CALL PAGER(11)	SORT 265
PRINT 8999	SORT 266
8999 FORMAT(/	SORT 267
1 /3(20X 17H XXXXX X X)	SORT 268
2 /3(20X 17HX X X X)	SORT 269
3 /3(20X 17HX X X X)	SORT 270
4 /3(20X 17HX X X)	SORT 271
5 /3(20X 17HX X X X)	SORT 272
6 /3(20X 17HX X X X)	SORT 273
7 /3(20X 17H XXXXX X X)	SORT 274
8 (/)	SORT 275
STOP	SORT 276
C ERRORS.	SORT 277
900 IF(.NOT.CONCAR) CALL DIVEST	SORT 278
CALL PAGER(2)	SORT 279
GO TO (901,901,903,904,905,	SORT 280
906,907,908,909),KE	SORT 281
901 PRINT 9001	SORT 282
9001 FORMAT(17H INPUT CARD ERROR, 111(1H*))	SORT 283
	SORT 284

GO TO 950		SORT 285
9003 PRINT 9003		SORT 286
9003 FORMAT(19H INCORRECT /FIELDS/,109(1H*))		SORT 287
GO TO 950		SORT 288
9004 PRINT 9004		SORT 289
9004 FORMAT(19H INCORRECT /SELECT/,109(1H*))		SORT 290
GO TO 950		SORT 291
9005 PRINT 9005		SORT 292
9005 FORMAT(27H EXTRANEIOUS INPUT STATEMENT,101(1H*))		SORT 293
GO TO 950		SORT 294
9006 PRINT 9006		SORT 295
9006 FORMAT(12H NO /FIELDS/,116(1H*))		SORT 296
GO TO 950		SORT 297
9007 PRINT 9007		SORT 298
9007 FORMAT(42H /ID/ TEXT DOES NOT MATCH ID ON INPUT TAPE,86(1H*))		SORT 299
GO TO 950		SORT 300
9008 PRINT 9008		SORT 301
9008 FORMAT(41H WRONG NUMBER OF FIELD LABELS IN /FIELDS/,87(1H*))		SORT 302
GO TO 950		SORT 303
9009 PRINT 9009, FLD(K)		SORT 304
9009 FORMAT(1X,A4,60H IS NOT A FIELD LABEL DEFINED FOR THE FILE ON THE		SORT 305
,INPUT TAPE,63(1H*))		SORT 306
950 PRINT 9050,KE		SORT 307
9050 FORMAT(118X,6HSORT I4)		SORT 308
STOP		SORT 309
C CIT(10000) CITATIONS CURRENTLY IN CORE, (SORT PHASE)		SORT 310
C CITIN(3,150) BUFFER FOR INPUT CITATION, SORT PHASE,		SORT 311
C CITM(3,150,2)CURRENT INPUT CITATIONS FROM MTAPE(1,MTA) AND		SORT 312
C MTAPE(2,MTA), MERGE PHASE,		SORT 313
C ENDIN TRUE IFF ALL CITATIONS HAVE BEEN READ FROM MTIN,		SORT 314
C FINAL TRUE IFF CURRENT MERGE PASS IS LAST ONE, MERGE PHASE,		SORT 315
C FLD(10) FIELD LABELS IN /FIELDS/,		SORT 316
C ID(13) ID FROM INPUT TAPE,		SORT 317
C IDA(13) ID FROM /ID/,		SORT 318
C IPASS PASS NUMBER, MERGE PHASE,		SORT 319
C ISTRNG STRING NUMBER IN CURRENT PASS, MERGE PHASE,		SORT 320
C JFIELD(10) ORDER OF FIELDS FOR SORTING, FROM /FIELDS/,		SORT 321
C KRANK(300) ORDERING OF THE CITATIONS, PRODUCED BY SHELL, THE		SORT 322
C CORE SORTING SUBROUTINE, (SORT PHASE)		SORT 323
C KT FIRST SUBSCRIPT OF MTAPE AND NSTRNG,		SORT 324
C LATE LAST CITATION NUMBER IN LSEL,		SORT 325
C LC INDEX IN CIT FOR NEXT CITATION, LC IS NEXT VALUE TO		SORT 326
C BE PUT INTO LCIT,		SORT 327
C LCIT(300) LCIT(L) IS THE SUBSCRIPT IN CIT OF THE BEGINNING OF		SORT 328
C ITS L-TH CITATION, (SORT PHASE)		SORT 329
C LCITIN NUMBER OF CITATIONS READ FROM MTIN, (SORT PHASE)		SORT 330
C LCITOU CURRENT COUNTER FOR NCITOU,		SORT 331
C LENCIT LENGTH OF CIT ARRAY, (10000)		SORT 332
C LENSTR NO. OF CITATIONS IN CURRENT OUTPUT STRING, MERGE PH,		SORT 333
C LSEL(100) LIST OF SELECTED CITATIONS, FROM /SELECT/,		SORT 334
C MT4 THE SECOND SUBSCRIPT OF MTAPE AND NSTRNG, SHOWS WHICH		SORT 335
C PAIR OF SCRATCH TAPES IS CURRENTLY THE INPUT PAIR		SORT 336
C DURING MERGE PHASE,		SORT 337
C MTB THE SECOND SUBSCRIPT OF MTAPE AND NSTRNG, SHOWING		SORT 338
C WHICH PAIR OF SCRATCH TAPES IS CURRENTLY THE OUTPUT		SORT 339
C PAIR DURING MERGE PHASE, (MTB=3-MTA)		SORT 340
C MTAPE(2,2) TAPE NUMBERS OF THE FOUR SCRATCH TAPES, MTAPE(1,1)		SORT 341

C		AND MTAPE(2,1) ARE USED IN SORT PHASE. ALL FOUR ARE	SORT 342
C		USEL IN MERGE PHASE,	SORT 343
C	MTIN	INPUT TAPE,	SORT 344
C	MTOUT	OUTPUT TAPE,	SORT 345
C	MXCIT	MAXIMUM NUMBER OF CITATIONS IN CIT. (SORT PHASE)	SORT 346
C		(300)	SORT 347
C	MXSEL	MAXIMUM LENGTH OF LSEL. (100)	SORT 348
C	MXWC	MAXIMUM NUMBER OF CELLS REQUIRED FOR A CITATION.	SORT 349
C	NC	NUMBER OF CELLS IN WHICH CURRENT CITATION IS STORED.	SORT 350
C	NCIT	NUMBER OF CITATIONS IN CIT. (SORT PHASE)	SORT 351
C	NCITOU	NUMBER OF CITATIONS ON MTOUT. (NUMBER OF CITATIONS IN	SORT 352
C		/SELECT/.)	SORT 353
C	NCUMPR	NUMBER OF COMPARISONS FOR ORDER, MERGE PHASE.	SORT 354
	NFLD	NUMBER OF FIELD LABELS IN /FIELDS/.	SORT 355
	NSTRNG(2,2)	NSTRNG(J,K) IS THE NO. OF STRINGS CURRENTLY ON	SORT 356
		MTAPE(J,K).	SORT 357
	NUMCIT(2,2)	NUMCIT(J,K) IS THE NO. OF CITATIONS CURRENTLY ON	SORT 358
		MTAPE(J,K).	SORT 359
	END		SORT 360

PROGRAM VOCAB(INPUT,OUTPUT,PUNCH,TAPE1,	VOCAB 2
TAPE50=INPUT,TAPE51=OUTPUT)	VOCAB 3
C VOCAB PROGRAM, PART OF THE FAMULUS SYSTEM,	VOCAB 4
C PROGRAMMED BY ROBERT M. RUSSELL, MARCH 1968.	VOCAB 5
COMMON /CORE/NCHW,BLANKS,BLANK	VOCAB 6
INTEGER BLANKS,BLANK	VOCAB 7
COMMON /FAM/SLASH,STAR,COMMA,ZERO,MXCHAR,MXCODES,MXWDES,MXCHW,	VOCAB 8
MXFREE,MXWFRE	VOCAB 9
INTEGER SLASH,STAR,COMMA,ZERO	VOCAB 10
COMMON /FIEL/FIELD(10),IFIELD(10),NFIELD,KDESCR	VOCAB 11
INTEGER FIELD	VOCAB 12
COMMON /INPT/CARD(10),KOLUM,HOLD,ENDCAR,HLACAR,CONCAR,ERRIN,NWC,	VOCAB 13
MXFL,MAXCAR,NCARIN,KHR,KIN,LIT,LITX,SYM(5),NUMB,NCSYM,CATLOG,	VOCAB 14
KARA,KARB	VOCAB 15
INTEGER CARD,SYM	VOCAB 16
LOGICAL HOLD,ENDCAR,HLACAR,CONCAR,ERRIN,CATLOG	VOCAB 17
COMMON /PAGING/IPAGE,LINES,MAXLIN,PAGHD(13)	VOCAB 18
COMMON /TEST/ITEST	VOCAB 19
COMMON /FREQU/ NFREQY(4,6500)	RJO 128
INTEGER SEMIC	RJO 129
LOGICAL KPUNCH	RJO 130
C	VOCAB 20
COMMON NDICT,NVOC,NTRIV,MXHASH,MXENT,MXDICT,WORD(5),	VOCAB 21
DICT(6750),LDICT(6501),HASH(9511)	VOCAB 22
INTEGER DICT,HASH,WORD,KRANK(6500)	VOCAB 23
EQUIVALENCE(KRANK,HASP)	VOCAB 24
INTEGER ID(13),IDA(13),LSEL(100),KFIELD(10),FLD(10)	VOCAB 25
INTEGER CIT(3,180),KODEW(64)	VOCAB 26
LOGICAL MEMBER	VOCAB 27
DATA LCIT,LCT,NFLD,LATE/4*0/	VOCAB 28
C INITIALIZE.	VOCAB 19
CALL USHR(IDA,KERR)	VOCAB 30
C---DEFINE SEMICOLON AS DELIMITER CODE	RJO 131
SEMIC = KHAR(1H),1)	RJO 132
DO 2 I = 1,4	RJO 133
DO 2 J = 1,6500	RJO 134
2 NFREQY(I,J) = 0	RJO 135
KPUNCH = .FALSE.	RJO 136
KI=1	VOCAB 31
IF(KERR.NE.0) GO TO 900	VOCAB 52
MXDICT=6750*NCHW	VOCAB 33
MXENT=6500	VOCAB 34
MXSEL=100	VOCAB 35
MXHASH=9511	VOCAB 36
NVOC=0	VOCAB 37
NTRIV=0	VOCAB 38
NDICT=0	VOCAB 35
MTIN=1	VOCAB 40
CALL REWARD(MTIN)	VOCAB 41
LDICT(1)=0	VOCAB 42
DO 5 K=1,64	VOCAB 43
KODEW(K)=0	VOCAB 44
IK=IDEAL(K-1)	VOCAB 45
IF(IK.EQ.2.OR.IK.EQ.3.OR.IK.EQ.8) KODEW(K)=1	VOCAB 46
C THE NEXT STATEMENT PUTS THE APOSTROPHE IN THE SET OF DESCRIPTOR CHARS.	VOCAB 47
IF(K-1.EQ.KHAR(1H),1)) KODEW(K)=1	VOCAB 48
C---ADD SEMICOLON AS DELIMITER IN SET OF DESCRIPTOR CODE CHARACTERS	RJO 137

```

        IF(K-1.EQ.KHAR(14;,1)) KODEW(K) = 1
    > CONTINUE
    CALL KORAN(5HKODEW,KODEW,64)
C NEXT STATEMENT.
    10 KE=13
        IF(ERRIN) GO TO 900
        IF(ITEST.NE.0) CALL PAGER(-4)
        CALL SETSYM
        IF(ENDCAP) GO TO 40
        CALL FABRIC(KON)
        KE=2
        IF(ERRIN) GO TO 900
        IF(KOLUM.NE.1) CALL DIVEST
        IF(KON.NE.2) GO TO 20
C /FIELD/
        KE=3
        IF(NFLD.NE.0) GO TO 900
        CALL GETLAB(FLD,1,10,NFLD,KERR)
        KE=4
        IF(KERR.GT.0) GO TO 900
        CALL TALMOD(3HFLD,FLD,0,NFLD*NCHW)
        GO TO 10
    20 IF(KON.NE.3) GO TO 30
C /SELECT/
        KE=5
        IF(LATE.NE.0) GO TO 900
        CALL GETNUM(LSEL,1,MXSEL,N,KERR)
        KE=6
        IF(KERR.GT.0) GO TO 900
        LATE=IABS(LSEL(N))
        CALL KORAN(4HSEL,LSEL,N+1)
        GO TO 10
    30 KE=7
        IF(KON.NE.11) GO TO 35
C /VOCABULARY/
        CALL NOAH(KERR)
        KE=8
        IF(KERR.GT.0) GO TO 900
        IF(ITEST.EQ.0) GO TO 10
        CALL PRN11C(2)
        CALL VSTAT(16,16HINPUT DICTIONARY)
        GO TO 10
    35 CONTINUE
C /PUNCH/ IF PUNCH CARD INCLUDED, DICTIONARY IS PUNCHED ON CARDS
        KPUNCH = .TRUE.
        GO TO 10
C ALL CARDS ARE READ. PREPARE TO RUN.
    49 KE=11
        IF(NDICT.EQ.0) GO TO 900
        CALL IOID(0,MTIN,ID,FIELD,IFIELD,NFIELD,KDESCR)
        CALL PAGER(-2)
        CALL IDPRIN(ID,FIELD,IFIELD,NFIELD,KDESCR,10,10HINPUT TAPE)
        CALL SIMILE(1,ID,0,MXFREE,IDA,0,MXFREE,LIKE)
        KE=9
        IF(LIKE.NE.1) GO TO 900
C SET LSEL AND LATE IF NO /SELECT/ STATEMENT.
        IF(LATE.NE.0) GO TO 50

```

```

RJO 138
VOCAB 49
VOCAB 50
VOCAB 51
VOCAB 52
VOCAB 53
VOCAB 54
VOCAB 55
VOCAB 56
VOCAB 57
VOCAB 58
VOCAB 59
VOCAB 60
VOCAB 61
VOCAB 62
VOCAB 63
VOCAB 64
VOCAB 65
VOCAB 66
VOCAB 67
VOCAB 68
VOCAB 69
VOCAB 70
VOCAB 71
VOCAB 72
VOCAB 73
VOCAB 74
VOCAB 75
VOCAB 76
VOCAB 77
VOCAB 78
VOCAB 79
VOCAB 80
RJO 13F
VOCAB 82
VOCAB 83
VOCAB 84
VOCAB 85
VOCAB 86
VOCAB 87
VOCAB 88
VOCAB 89
RJO 141
RJO 142
RJO 143
VOCAB 90
VOCAB 91
VOCAB 92
VOCAB 93
VOCAB 94
VOCAB 95
VOCAB 96
VOCAB 97
VOCAB 98
VOCAB 99
VOCAB100

```

```

LATE=100000
LSEL(1)=1
LSEL(2)=-LATE
LSEL(3)=0
C SET FLD AND NFLD IF NO /FIELDS/ STATEMENT.
50 IF(NFLD.NE.0) GO TO 70
K1=12
IF(KDESCR.EQ.0) GO TO 90n
NFLD=1
FLD(1)=FIELD(KDESCR)
C CONVERT FLD LABELS TO KFIELD INDICATORS.
70 CALL MOVE(0,KFIELD,-NFLD)
DO 80 K=1,NFLD
L=LOOK(FIELD,NFIELD,FLD(K))
KF=10
IF(L.GT.NFIELD) GO TO 900
80 KFIELD(L)=1
CALL KORAN(6HKFIELD,KFIELD,NFIELD)
C READ NEXT CITATION FROM INPUT TAPE.
100 IF(LCIT.GE.LATE) GO TO 150
IF(ITEST.NE.0) CALL PAGER(-5)
CALL KORAN(4HLCIT,LCIT+1,1)
CALL IOTAPE(0,MTIN,CIT,LAST)
IF(LAST.EQ.1) GO TO 150
LCIT=LCIT+1
IF(.NOT.MEMBER(LSEL,MXSEL,LCIT)) GO TO 100
C SCAN EACH FIELD OF CIATION.
DO 130 KF=1,NFIELD
IF(KFIELD(KF).EQ.0) GO TO 130
NC=CIT(1,KF)
IF(NC.EQ.0) GO TO 130
IF(ITEST.NE.0) CALL PAGER(-2)
CALL TALMUD(5HFIELD,FIELD(KF),0,MXFL)
LC=CIT(2,KF)
KH=BLANK
IF(KF.EQ.KDESCR) KH=COMMA
C PICK OUT NEXT WORD OR PHRASE FROM CIT.
110 IF(KF.EQ.KDESCR) GO TO 113
CALL SCAN(0,KODEW,CIT,LC,NC,N,KH)
GO TO 114
C---CHANGE DELIMITER IN DESCRIPTOR FIELD TO SEMICOLON - RJ OLSON 07/73
112 CALL SCOUT(0,1,SEMIC,CIT,LC,NC,NN)
CALL SCOUT(1,0,BLANK,CIT,LC,NN,M)
N=NN-M
114 NIP=MAX0(N,0)
N=MIN0(N,MXCDFS)
IF(N.EQ.0) GO TO 120
CALL SCAT(CIT,LC,N,DICT,LDICT,NDICT,HASH,MXHASH,JDICT,JHASH)
CALL SPOT(JDICT,JHASH,N,NN,4HIIII,24HJDICT JHASH N NN )
C---COUNT FREQUENCY OF OCCURENCE AND CITATION NUMBERS OF FIRST THREE
C---ADD BY HJ OLSON JULY, 1973 (NFREQY(4,4500) = NEW ARRAY)
NFREQY(1,JDICT) = NFREQY(1,JDICT) + 1
NNF = NFREQY(1,JDICT) + 1
IF (NNF .GT. 4) GO TO 116
NFREQY(NNF,JDICT) = LCIT
116 CONTINUE
IF(JDICT.LE.NDICT) GO TO 120

```

```

VOCAB101
VOCAB102
VOCAB103
VOCAB104
VOCAB105
VOCAB106
VOCAB107
VOCAB108
VOCAB109
VOCAB110
VOCAB111
VOCAB112
VOCAB113
VOCAB114
VOCAB115
VOCAB116
VOCAB117
VOCAB118
VOCAB119
VOCAB120
VOCAB121
VOCAB122
VOCAB123
VOCAB124
VOCAB125
VOCAB126
VOCAB127
VOCAB128
VOCAB129
VOCAB130
VOCAB131
VOCAB132
VOCAB133
VOCAB134
VOCAB135
VOCAB136
VOCAB137
VOCAB138
VOCAB139
VOCAB140
HJO 144
HJO 145
VOCAB142
VOCAB143
VOCAB144
VOCAB145
VOCAB146
VOCAB147
VOCAB148
RJO 146
RJO 147
RJO 148
RJO 149
HJO 150
RJO 151
RJO 152
VOCAB149

```

```

NDICT=NDICT+1
IF (NDICT.GT.MXENT) GO TO 140
LD=LDICT(NDICT)
LDICT(NDICT+1)=LD+N
IF (LD+N.GT.MXDICT) GO TO 140
CALL CHOP(CIT,LC,DICT,LD,N)
CALL TALMOD(4HDICT,DICT,LD,N)
HASH(JHASH)=NDICT
120 LC=LC+NIP+1
NC=NC-NIP-1
CALL SCAN(1,KODEW,CIT,LC,NC,M,KH)
LC=LC+M
NC=NC-M
IF (NC.GT.0) GO TO 110
130 CONTINUE
C FINISHED SCANNING CITATION.
LCT=LCIT
IF (ITEST.NE.0) CALL PRNDIC(2)
GO TO 100
C ERROR MESSAGES FOR FULL DICTIONARY.
140 CALL PAGER(5)
NDICT=NDICT-1
PRINT 1000,NDICT,LDICT(NDICT)
1000 FORMAT(/20H DICTIONARY IS FULL./20H NUMBER OF ENTRIES =4/
, 23H NUMBER OF CHARACTERS =16/1X)
C PREPARE FOR OUTPUT.
150 CALL REWARD(MTIN)
AVOC=NDICT-NTRIV
IF (ITEST.NE.0) CALL PRNDIC(2)
CALL SHELLV(DICT,KRANK,NDICT)
C-----CALL SHELL CHANGED HERE TO SHELLV BY JOHN R. TADLOCK NOV. 1975.
CALL KORAN(5HKRANK,KRANK,NDICT)
C PUNCH DICTIONARY.
IF (KPUNCH) CALL PUNDIC
C PRINT TRIVIAL ENTRIES.
CALL PRNDIC(0)
C PRINT VOCABULARY ENTRIES.
CALL PRNDIC(1)
C MISCELLANEOUS PRINTOUT.
CALL PAGER(2)
PRINT 2000,LCT
2000 FORMAT(32H THE LAST CITATION INSPECTED WAS,15/1X)
CALL VSTAT(17,17HOUTPUT DICTIONARY)
CALL PAGER(11)
PRINT 8999
8999 FORMAT(/
1 /3(20X 17H XXXXX X X)
2 /3(20X 17HX X X X)
5 /3(20X 17HX X X X)
4 /3(20X 17HX X X)
5 /3(20X 17HX X X X)
6 /3(20X 17HX X X X)
7 /3(20X 17h XXXXX X X)
8 //)
STOP
C ERROR MESSAGES,
900 IF (KOLUM.GT.1) CALL DIVEST

```

VOCAB150
VOCAB151
VOCAB152
VOCAB153
VOCAB154
VOCAB155
VOCAB156
VOCAB157
VOCAB158
VOCAB159
VOCAB160
VOCAB161
VOCAB162
VOCAB163
VOCAB164
VOCAB165
VOCAB166
VOCAB167
VOCAB168
VOCAB169
VOCAB170
VOCAB171
VOCAB172
VOCAB173
VOCAB174
VOCAB175
VOCAB176
VOCAB177
VOCAB178
JRT 179
JRT
VOCAB180
VOCAB181
RJO 153
VOCAB183
VOCAB184
VOCAB185
VOCAB186
VOCAB187
VOCAB188
VOCAB189
VOCAB190
VOCAB191
VOCAB192
VOCAB193
VOCAB194
VOCAB195
VOCAB196
VOCAB197
VOCAB198
VOCAB199
VOCAB200
VOCAB201
VOCAB202
VOCAB203
VOCAB204
VOCAB205

CALL PAGER(2)	VOCAB206
GO TO (901,901,903,904,903,	VOCAB207
904,903,905,906,907,	VOCAB208
901,902,901),XE	VOCAB209
901 PRINT 9001	VOCAB210
9001 FORMAT(12H INPUT ERROR,116(1H*))	VOCAB211
GO TO 950	VOCAB212
902 PRINT 9002	VOCAB213
9002 FORMAT(30H YOU NEED A /FIELDS/ STATEMENT,98(1H*))	VOCAB214
GO TO 950	VOCAB215
903 PRINT 9003	VOCAB216
9003 FORMAT(21H EXTRANEIOUS STATEMENT,107(1H*))	VOCAB217
GO TO 950	VOCAB218
904 PRINT 9004	VOCAB219
9004 FORMAT(25H ERROR IN ABOVE STATEMENT,103(1H*))	VOCAB220
GO TO 950	VOCAB221
905 PRINT 9005	VOCAB222
9005 FORMAT(27H DICTIONARY LIMITS EXCFEDED,101(1H*))	VOCAB223
GO TO 950	VOCAB224
906 PRINT 9006	VOCAB225
9006 FORMAT(28H /ID/ DOES NOT MATCH TAPE ID,100(1H*))	VOCAB226
GO TO 950	VOCAB227
907 PRINT 9007	VOCAB228
9007 FORMAT(35H INCORRECT FIELD LABELS IN /FIELDS/,93(1H*))	VOCAB229
950 PRINT 9050,KE	VOCAB230
9050 FORMAT(118X,6HVOCAB 14)	VOCAB231
STOP	VOCAB232
C FORTRAN VARIABLES,.,.	VOCAB233
C CIT(3,150) CURRENT CITATION,	VOCAB234
C DICT(5625) VOCABULARY AND TRIVIAL WORDS (AVERAGE, 10 CHARACTERS	VOCAB235
C PER WORD),	VOCAB236
C FLD(10) TEMPORARY STORAGE FOR FIELD LABELS FROM /FIELDS/.	VOCAB237
C HASH(5501) HASH CODE TABLE, HASH(LH)=0 INDICATES EMPTY POSITION,	VOCAB238
C NONZERO VALUE IS THE INDEX IN LDICT POINTING TO THE	VOCAB239
C DICT ENTRY HAVING HASH TABLE POSITION LH,	VOCAB240
C IT(13) TEXT FROM TAPE I.D.	VOCAB241
C IDA(13) TEXT FROM /ID/.	VOCAB242
C KFIELD(10) FIELDS TO COLLECT ENTRIES FROM, IF KFIELD(J)=1,	VOCAB243
C COLLECT FROM FIELD J, IF KFIELD(J)=0, DO NOT COLLECT	VOCAB244
C FROM FIELD J,	VOCAB245
C KRANK(4500) INDICATORS OF ALPHABETICAL ORDER OF DICT ENTRIES,	VOCAB246
C (SHARES MEMORY WITH HASH)	VOCAB247
C LATE LAST CITATION THAT NEEDS TO BE INSPECTED, I.E., THE	VOCAB248
C LAST ONE IN /SELECT/,	VOCAB249
C LCIT CURRENT CITATION NUMBER,	VOCAB250
C LCT VALUE OF LCIT FOR MOST RECENT CITATION THAT HAS BEEN	VOCAB251
C INSPECTED SO FAR IN RUN,	VOCAB252
C LDICT(4501) LDICT(J) POINTS TO THE BEGINNING OF THE J-TH ENTRY IN	VOCAB253
C DICT. A NEGATIVE SIGN ON LDICT(J) INDICATES ENTRY IS	VOCAB254
C A TRIVIAL WORD, LDICT(NDICT+1) POINTS TO THE	VOCAB255
C CHARACTER POSITION FOLLOWING THE LAST CHARACTER IN	VOCAB256
C DICT, THUS IT CAN BE USED TO CALCULATE THE LENGTH	VOCAB257
C OF EACH DICT SUBSTRING,	VOCAB258
C CITATION LIST FROM /SELECT/,	VOCAB259
C LSEL(100)	VOCAB260
C MTIN INPUT TAPE,	VOCAB261
C MXDICT MAXIMUM NUMBER OF CHARACTERS THAT DICT CAN HOLD,	VOCAB262
C MXENT MAXIMUM NUMBER OF ENTRIES IN DICTIONARY, NAMELY,	VOCAB262

	MAXIYUM LENGTH OF LDICT LESS ONE,	VOCAB263
MXHASH	LENGTH OF HASH ARRAY. MXHASH MUST BE A PRIME NUMOER	VOCAB264
	IN THE NEIGHBORHOOD OF $1.5 * MXENT$, ACCORDING TO METHOD	VOCAB265
	OF W.D. MAURER, CACM, JAN. 1968,	VOCAB266
C	MXSEL	VOCAB267
C	NDICT	VOCAB268
C	NDICT=NVOC+NTRIV	VOCAB269
C	NFLD	VOCAB270
C	NTRIV	VOCAB271
	NUMBER OF ENTRIES WHICH ARE TRIVIAL WORDS,	VOCAB272
	NUMBER OF ENTRIES WHICH ARE VOCABULARY ENTRIES,	VOCAB273
	WORD(5)	VOCAB274
	END	

SUBROUTINE ADAM(TEXT1,LCH1,NCH1,TEXT2,LCH2,NCH2,PATREP,LPAT,LREP,	AUAM	2
NCPAT,NCREP,NPAT,MAYBE,DIFF,NAR)	ADAM	3
C THIS SUBROUTINE HANDLES THE DETAILS OF THE REPLACE STATEMENT IN	ADAM	4
C FAMULUS EDIT, IT OPERATES ON A SINGLE FIELD, MAKING A REPLACEMENT	ADAM	5
C WHEREVER A GIVEN PATTERN MATCHES THE TEXT,	ADAM	6
C THE TEXT OF THE FIELD IS THUS TRANSFERRED FROM TEXT1 SUBSTRING TO	ADAM	7
C TEXT2 SUBSTRING AND THE NUMBER OF CHARACTERS IN THE LATTER IS SET AS	ADAM	8
C NCH2.	AUAM	9
C A PATTERN WITH ZERO NUMBER OF CHARACTERS IS OK, IT HAS THE EFFECT OF	ADAM	10
C ADDING MATERIAL ONTO THE END, AND, IF THE TEXT1 FIELD IS ALSO VOID	ADAM	11
C (NCH1=0), THE EFFECT IS TO CREATE A FIELD,	ADAM	12
C A REPLACEMENT WITH ZERO NUMBER OF CHARACTERS DELETES TEXT, IN CASE THE	ADAM	13
C COKHESPONUNG PATTERN OR PATTERNS COVERS THE ENTIRE FIELD, THE EFFECT	ADAM	14
C IS TO DELETE A FIELD,	ACAM	15
C IN THE RESULTING TEXT2 FIELD, ANY LEADING OR TRAILING BLANKS ARE	ADAM	16
C ELIMINATED,	ADAM	17
C IN CASE ANY REPLACEMENT IS MADE, DIFF IS SET TO TRUE, OTHERWISE, DIFF	ADAM	18
C IS UNALTERED,	ADAM	19
C LCH1 POINTS TO THE START OF THE FIELD IN TEXT1.	ADAM	20
C LCH1 MUST HE A MULTIPLE OF NCHW.	ADAM	21
C LCH2 IS SIMILARLY DEFINED, FOR TEXT2.	ADAM	22
C NCH1 IS THE NUMBER OF CHARACTERS IN THE TEXT1 FIELD TO BE PROCESSED.	ADAM	23
C NCH2 IS THE NUMBER OF CHARACTERS IN THE RESULTING TEXT2 FIELD,	ADAM	24
C PATREP IS A CHARACTER STRING CONTAINING THE TEXT OF PATTERNS AND	ADAM	25
C REPLACEMENTS,	ADAM	26
C NCPAT(J) IS THE NUMBER OF CHARACTERS IN PATTERN J,	ALAY	27
C NCREP(J) IS THE NUMBER OF CHARACTERS IN REPLACEMENT J,	ACAM	28
C LPAT(J) IS A POINTER TO PATREP FOR THE FIRST CHARACTER IN PATTERN J,	ADAM	29
C THE CHARACTER POSITION IS LPAT(J)+1.	ADAM	30
C LREP(J) IS A POINTER TO PATREP FOR THE FIRST CHARACTER IN REPLACEMENT	ADAM	31
C J, THE CHARACTER POSITON IS LREP(J)+1.	ADAM	32
C NPAT IS THE NUMBER OF PATTERNS (AND THE NUMBER OF REPLACEMENTS),	ADAM	33
C IF NPAT=0, THE FIELD IS COPIED INTO TEXT2 WITHOUT CHANGE,	ADAM	34
C NAR IS THE NUMBER OF CHARACTERS IN A LEADING ARTICLE, IF ANY, IN	ADAM	35
C RESULT FIELD, (ARTICLES ARE A, AN, THE),	ADAM	36
C MAYBE(J)=1 IF CHARACTER CODE J-1 IS THE FIRST CHARACTER OF ONE OF THE	ADAM	37
C PATTERNS, =0 OTHERWISE. THIS IS FOR SPEED,	ADAM	38
COMMON /CORE/NCHW,BLANKS,BLANK	ADAM	39
INTEGER BLANKS,BLANK	ADAM	40
COMMON /FAM/SLASH,STAR,COMMA,ZERO,MXCHAR,MXCODES,MXWDES,MXCHW,	ADAM	41
MXFREE,MXWFRE	ADAM	42
INTEGER SLASH,STAR,COMMA,ZERO	ADAM	43
COMMON /TEST/ITEST	ADAM	44
INTEGER TEXT1(2),TEXT2(2),PATREP(2),LPAT(2),LREP(2),NCPAT(2),	ADAM	45
NCREP(2),MAYBE(64)	ADAM	46
LOGICAL DIFF	ADAM	47
KUT(N)=(N+NCHW-1)/NCHW	ADAM	48
C INITIALIZE,	ADAM	49
LC1=LCH1	ADAM	50
LC2=LCH2	ADAM	51
LEFT1=NCH1	ADAM	52
LEFT2=MXCHAR-LC2	ADAM	53
IF(ITEST,EQ,0) GO TO 8	ADAM	54
CALL BUTLER	ADAM	55
CALL SPOT(LC1,LEFT1,LC2,LEFT2,4HIIII,24HLC1 LEFT1 LC2 LEFT2)	ADAM	56
CALL SPOT(NPAT,DIFF,0,0,4HIIII,24HNPAT DIFF)	ADAM	57
CALL TALMUD(5HTEXT1,TEXT1,LCH1,NCH1)	ADAM	58

CALL KORAN(5H MAYRE, MAYRE, 64)	ADAM	59
IF(NPAT, LE, 0) GO TO 6	ADAM	60
DO 4 JP=1, NPAT	ADAM	61
CALL TALMUD(3HPAT, PATREP, LPAT(JP), NCPAT(JP))	ADAM	62
4 CALL TALMUD(3HREP, PATREP, LREP(JP), NCREP(JP))	ADAM	63
6 CALL PAGER(-1)	ADAM	64
6 CONTINUE	ADAM	65
IF(NPAT, LE, 0) GO TO 90	ADAM	66
M=0	ADAM	67
C SCAN FOR START OF NEXT POSSIBLE PATTERN.	ADAM	68
10 CALL SCAN(1, MAYRE, TEXT1, LC1+M, LEFT1-M, NC, KH)	ADAM	69
MORE=MIND(NC+M, LEFT2)	ADAM	70
C COPY CHARACTERS THAT CANNOT BE FIRST CHARACTER OF A PATTERN.	ADAM	71
CALL CHOP(TEXT1, LC1, TEXT2, LC2, MORE)	ADAM	72
CALL TALMUD(5HTEXT1, TEXT1, LC1, MORE)	ADAM	73
LC1=LC1+MORE	ADAM	74
LC2=LC2+MORE	ADAM	75
LEFT1=LEFT1-MORE	ADAM	76
LEFT2=LEFT2-MORE	ADAM	77
IF(LEFT1, LE, 0, OR, LEFT2, LE, 0) GO TO 70	ADAM	78
C COMPARE EACH PATTERN WITH TEXT1.	ADAM	79
DO 40 JP=1, NPAT	ADAM	80
CALL SIMILE(0, PATREP, LPAT(JP), NCPAT(JP), TEXT1, LC1, LEFT1, LIKE)	ASAP	E 1
IF(LIKE, EQ, 1) GO TO 50	ADAM	82
40 CONTINUE	ADAM	83
C NO PATTERNS MATCHED.	ADAM	84
M=1	ADAM	85
GO TO 10	ADAM	86
C A PATTERN MATCHED. INSERT REPLACEMENT.	ADAM	87
50 MORE=MIND(NCREP(JP), LEFT2)	ADAM	88
CALL CHOP(PATREP, LREP(JP), TEXT2, LC2, MORE)	ADAM	89
CALL SPOT(JP, MORE, NCPAT(JP), NCREP(JP), 4HIIII,	ADAM	90
, 24HJP MORE NCPAT NCREP)	ADAM	91
CALL TALMUD(3HREP, PATREP, LREP(JP), MORE)	ADAM	92
CALL TALMUD(5HTEXT2, TEXT2, LC2, MORE)	ADAM	93
DIFF=, TRUE.	ADAM	94
M=0	ADAM	95
LC1=LC1+NCPAT(JP)	ADAM	96
LC2=LC2+MORE	ADAM	97
LEFT1=LEFT1-NCPAT(JP)	ADAM	98
LEFT2=LEFT2-MORE	ADAM	99
IF(LEFT1, GT, 0, AND, LEFT2, GT, 0) GO TO 10	ADAM	100
C INSERT MATERIAL WHICH REPLACES AN EMPTY PATTERN.	ADAM	101
70 CONTINUE	ADAM	102
CALL SPOT(LEFT1, LEFT2, 0, 0, 4HIIII, 24HLEFT1 LEFT2	ADAM	103
DO 80 JP=1, NPAT	ADAM	114
IF(NCPAT(JP), GT, 0) GO TO 80	ADAM	105
NRP=MIND(NCREP(JP), LEFT2)	ADAM	106
CALL CHOP(PATREP, LREP(JP), TEXT2, LC2, NRP)	ADAM	107
CALL TALMUD(3HREP, PATREP, LREP(JP), NRP)	ADAM	108
CALL TALMUD(5HTEXT2, TEXT2, LC2, NRP)	ADAM	109
DIFF=, TRUE.	ADAM	110
LC2=LC2+NRP	ADAM	111
LEFT2=LEFT2-NRP	ADAM	112
80 CONTINUE	ADAM	113
NCH2=LC2-LCH2	ADAM	114
CALL JUST(TEXT2, LCH2, NCH2, NSUP, NADD)	ADAM	115

NCH2=NCH2-NSUR	ADAM 116
GO TO 100	ADAM 117
C COPY FIELD INTACT BECAUSE NPAT=0.	ADAM 118
90 J1=KUT(LC1)+1	ADAM 113
J2=KUT(LC2)+1	ADAM 120
CALL MOVE(TEXT1(J1),TEXT2(J2),KUT(NCH1))	ADAM 121
NCH2=NCH1	ADAM 122
C RETURN	ADAM 123
100 CALL ARTY(TEXT2,LCH2,NCH2,NAR)	ADAM 124
CALL SPOT(NCH2,NAR,DIFF,0,4HIIL1,24HNCH2 NAR DIFF)	ADAM 125
CALL TALMUD(5HTEXT2,TEXT2,LCH2,NCH2)	AGAM 126
RETURN	ADAM 127
C TEMPORARY FORTRAN VARIABLES ARE ...	ADAM 128
C LC1 IS THE UPDATED LCH1, AS TEXT1 IS SCANNED.	ADAM 129
C LC2 IS THE UPDATED LCH2, AS TFX2 IS FILLED,	ADAM 130
C JP IS THE INDEX OF THE PATTERN CURRENTLY BEING PROCESSED.	ADAM 131
C LEFT1 IS THE NUMBER OF CHARACTERS IN TEXT1 FIELD THAT ARE STILL TO BE	ADAM 132
C PROCESSED (SCANNED).	ADAM 133
C LEFT2 IS THE MAXIMUM NUMBER OF CHARACTERS THAT TEXT2 STILL HAS ROOM	ADAM 134
C FOR,	ADAM 135
END	ADAM 136

SUBROUTINE ARTY(X,LX,NX,NRT)	ARTY	2
C GIVEN A SUBSTRING OF A CHARACTER STRING X, ARTY DETERMINES WHETHER IT	ARTY	3
C STARTS WITH AN ARTICLE (A, AN, THE). THE SUBSTRING IS NX CHARS	ARTY	4
C BEGINNING AT LX+1.	ARTY	5
C IF NO LEADING ARTICLE, SETS NRT=0. IF THERE IS ONE, SETS NRT EQUAL TO	ARTY	6
C THE NUMBER OF CHARACTERS IN THE ARTICLE, INCLUDING THE BLANK THAT	ARTY	7
C FOLLOWS IT. I.E., HOW MANY TO SKIP WHEN YOU WANT TO IGNORE ARTICLES,	ARTY	8
INTEGER X(2),ART(3),NART(3)	ARTY	9
DATA (ART(J),J=1,3)/2HA , 3HAN , 4HTHE /	ARTY	10
DATA (NART(J),J=1,3)/2,3,4/	ARTY	11
IF(NX.LT.1) GO TO 20	ARTY	12
DO 10 J=1,3	ARTY	13
NRT=NART(J)	ARTY	14
CALL SIMILE(0,ART(J),0,NRT,X,LX,NX,K)	ARTY	15
IF(K.EQ.1) RETURN	ARTY	16
10 CONTINUE	ARTY	17
20 NRT=0	ARTY	18
RETURN	ARTY	19
END	ARTY	20

LOGICAL FUNCTION ASCENI (X,J,K)	ASCENI 2
C USED IN INDEX PROGRAM TO COMPARE TWO INDEX ENTRIES FOR GRDER IN INDEX,	ASCENI 3
C CALLED BY SHELL.	ASCENI 4
C J AND K ARE THE INDICES OF THE TWO ENTRIES, AN ENTRY CONSISTS OF A	ASCENI 5
C PAIR OF INTEGERS, ONE EACH FROM LOCCIT() AND LOCDES() ARRAYS,	ASCENI 6
C PRECEDENCE IS DETERMINED AS FOLLOWS. LET JD=LOCDES(J) AND KD=LOCDES(K)	ASCENI 7
C , THE JD-TH DESCRIPTOR IN DICT IS COMPARED WITH THE KD-TH ONE, AND	ASCENI 8
C LOCCIT(J) IS COMPARED WITH LOCCIT(K). ASCEND IS SET AS FOLLOWS ...	ASCENI 9
C JD PRECEDES KD, ASCEND=TRUE	ASCENI10
C JD FOLLOWS KD, ASCEND=FALSE	ASCENI11
C JD=KD AND LOCCIT(J).LE,LOCCIT(K),ASCEND=TRUE	ASCENI12
C JD=KD AND LOCCIT(J).GT,LOCCIT(K), ASCEND=FALSE	ASCENI13
C X IS NOT USED.	ASCENI14
C	ASCENI15
COMMON CIT(3,180), DICT(6500),LDICT(4001), HASH(9511),	JRT
+ LOCCIT(8000), LOCDES(8000)	ASCENI17
INTEGER CIT,DICT,HASH	ASCENI18
C	ASCENI19
INTEGER X(1)	ASCENI20
JD=LOCDES(J)	ASCENI21
LJ=LDICT(JD)	ASCENI22
NJ=LDICT(JD+1)-LJ	ASCENI23
KD=LOCDES(K)	ASCENI24
LK=LDICT(KD)	ASCENI25
NK=LDICT(KD+1)-LK	ASCENI26
LEX=LEXICO(DICT,LJ,NJ,DICT,LK,NK)	ASCENI27
IF(LEX)30,10,20	ASCENI28
10 IF(LOCCIT(J).LE,LOCCIT(K)) GO TO 30	ASCENI29
20 ASCENI = ,FALSE,	ASCENI30
GO TO 100	ASCENI31
30 ASCENI = ,TRUE,	ASCENI32
100 RETURN	ASCENI33
END	ASCENI34

LOGICAL FUNCTION ASCENS (X, J, K)	JRT
C COMPARES TWO CITATIONS FOR ALPHABETICAL ORDER,	ASCENS 3
C---- CALLED BY "SHELLS"	JRT
C USED BY SHELL SORT ROUTINE TO COMPARE TWO ITEMS FOR PRECEDENCE IN	ASCENS 4
C SORTING, J AND K ARE THE INDICES OF THE TWO CITATIONS BEING SORTED,	ASCENS 5
C X IS NOT USE).	ASCENS 6
C ASCEND=TRUE IF J PRECEDES K OR THEY ARE TIED,	ASCENS 7
C ASCEND=FALSE IF K PRECEDES J,	ASCENS 8
COMMON LOCCIT(300),CIT(10000)	ASCENS 9
INTEGER CIT	ASCENS10
LA=LOCCIT(J)	ASCENS11
LB=LOCCIT(K)	ASCENS12
ASCENS = LEADER (CIT (LA), CIT (LB)) ,NE. 1	JRT 17
RETURN	ASCENS14
END	ASCENS15

```

      LOGICAL FUNCTION ASCENV ( X, J, K )
C USED IN VOCAB TO COMPARE TWO DICTIONARY ENTRIES FOR ALPHABETICAL
C ORDER.
C J AND K ARE THE INDICES OF THE TWO ENTRIES.
C ALL TRIVIAL ENTRIES PRECEDE ALL VOCABULARY ENTRIES.
C ASCEND=TRUE IF J PRECEDES K OR IF THEY ARE IDENTICAL.
C ASCENT=FALSE IF K PRECEDES J,
C X IS NOT USED,
      COMMON NDICT,NVOC,NTRIV,MXHASH,MXENT,MXDICT,WORD(5),
      .   DICT(6750),LDICT(6501),HASH(9511)
      INTEGER DICT,HASH,WORD,KRANK(6500)
      EQUIVALENCE(KRANK,HASH)
      LJ=IABS(LDICT(J))
      LK=IABS(LDICT(K))
      NJ=IABS(LDICT(J+1))-LJ
      NK=IABS(LDICT(K+1))-LK
      IF(ISIGN(1,LDICT(J)).EQ.ISIGN(1,LDICT(K))) GO TO 10
      ASCENV =  ISIGN ( 1, LDICT (J) ) .LT. 0
      GO TO 20
10  ASCENV =  LEXICO ( DICT, LJ, NJ, DICT, LK, NK ) .LE. 0
20  RETURN
      END

```

```

JRT  18
ASCENV 3
ASCENV 4
ASCENV 4
ASCENV 6
ASCENV 7
ASCENV 8
ASCENV 9
ASCENV10
ASCENV11
ASCENV12
ASCENV13
ASCENV14
ASCENV15
ASCENV16
ASCENV17
ASCENV18
JRT  24
ASCENV20
JHT  25
ASCENV22
ASCENV23

```

SUBROUTINE BOOLE(IFFY)	BOOLE 2
C EVALUATES THE BOOLEAN FORMULA IN LUKE USING VALUES IN LOCATE ARRAY.	BOOLE 3
C SETS IFFY TO INDICATE THE TRUTH VALUE OF THE FORMULA, NAMELY,	BOOLE 4
C IFFY=0, THE VALUE IS FALSE.	BOOLE 5
C IFFY=1, THE VALUE IS TRUE.	BOOLE 6
C THE FORMULA IS A POLISH STRING IN LUKE.	BOOLE 7
COMMON TEXT(250),LTEXT(50),NOTEXT(50),XTXT,	BOOLE 8
, JTERM(60),MODELD(60),LOCATE(60),NTERM,PAYBE(64),KODET(64),	BOOLE 9
, KUK(150),NKUK,LUKE(150),NLUKE,LIST(150),NLIST,	BOOLE 10
, KSERCH(10),NSERCH,KFLD(10),MXTEXT,MXCTXT,MXTERM,MXLUKE	BOOLE 11
CALL BUTLER	BOOLE 12
LS=0	BOOLE 13
30 40 LU=1,NLUKE	BOOLE 14
LK=LUKE(LU)	BOOLE 15
IF(LK,EQ,0) GO TO 50	BOOLE 16
IF(LK,LT,0) GO TO 5	BOOLE 17
LS=LS+1	BOOLE 18
LIST(LS)=LOCATE(LK)	BOOLE 19
GO TO 40	BOOLE 20
5 LOP=-LK	BOOLE 21
IF(LOP,LE,2) LS=LS-1	BOOLE 22
GO TO (10,20,30), LOP	BOOLE 23
C OR,	BOOLE 24
10 LIST(LS)=MAX0(LIST(LS),LIST(LS+1))	BOOLE 25
GO TO 40	BOOLE 26
C AND,	BOOLE 27
20 LIST(LS)=MIN0(LIST(LS),LIST(LS+1))	BOOLE 28
GO TO 40	BOOLE 29
C NOT,	BOOLE 30
30 LIST(LS)=1-LIST(LS)	BOOLE 31
40 CONTINUE	BOOLE 32
50 IFFY=LIST(1)	BOOLE 33
RETURN	BOOLE 34
END	BOOLE 35

SUBROUTINE BUTLER	BUTLER 2
C PRINTS A MESSAGE GIVING THE NAME, SAY A, OF THE SUBROUTINE WHICH HAS	BUTLER 3
C CALLED BUTLER AND ALSO THE NAME, SAY B, OF THE SUBROUTINE WHICH	BUTLER 4
C CALLED A, THE INTERNAL FORMULA NUMBER IN 3 OF THE CALL TO A IS ALSO	BUTLER 5
C PRINTED. THIS IS USED ONLY FOR DEBUGGING, SO IF /TEST/ITEST#0,	BUTLER 6
C NOTHING IS PRINTED.	BUTLER 7
C PROGRAMMED BY HOBERT M. RUSSELL, JANUARY, 1968.	BUTLER 8
COMMON /TEST/ITEST	BUTLER 9
IF(ITEST,EQ,0) GO TO 10	BUTLER10
CALL FERRET(SUBA,IFNA,SURB,IFNB)	BUTLER11
CALL PAGER(1)	BUTLER12
PRINT 1000, SUBB,IFNB,SUBA	BUTLER13
10 RETURN	BUTLER14
1000 FORMAT(1X,10(1H.),1X,A7,05,11H IS CALLING,1X,A7,1X,10(1H.))	BUTLER15
END	BUTLER16

SUBROUTINE CARCIT(CIT,CRD,NCRD,ERR)	CARCIT 2
C READS ONE CITATION FROM CARDS, STORING IT IN THE ARRAY CIT IN STANDARD	CARCIT 3
C FAMULUS FORMAT.	CARCIT 4
C CRD IS A TEMPORARY STORAGE ARRAY IN WHICH CARCIT STORES THE CITATION	CARCIT 5
C IN CARD IMAGE FORM, IT MUST BE BIG ENOUGH FOR 63 CARDS.	CARCIT 6
C ERR IS SET TRUE IF THE CARDS CONTAIN AN ERROR, FALSE OTHERWISE.	CARCIT 7
C NCRD IS THE NUMBER OF CARDS IN THE CITATION. NCRD=0 IF END OF INPUT.	CARCIT 8
COMMON /CORE/NCHW,BLANKS,BLANK	CARCIT 9
INTEGER BLANKS,BLANK	CARCIT10
COMMON /FAM/SLASH,STAR,COMMA,ZERO,MXCHAR,MXCDER,MXWDES,MXCHW,	CARCIT11
MXFREE,MXWFRE	CARCIT12
INTEGER SLASH,STAR,COMMA,ZERO	CARCIT13
COMMON /FIEL/FIELD(10),IFIELD(10),NFIELD,KDESCR	CARCIT14
INTEGER FIELD	CARCIT15
COMMON /INPT/CARD(10),KOLUM,HOLD,ENDCAR,BLACAR,CONCAR,ERRIN,NWC,	CARCIT16
MXFL,MXCAR,NCARIN,KHR,KIN,LIT,LITX,SYM(5),NUMB,NCSYM,CATLOG,	CARCIT17
KARA,KARB	CARCIT18
INTEGER CARD,SYM	CARCIT19
LOGICAL HOLD,ENDCAR,BLACAR,CONCAR,ERRIN,CATLOG	CARCIT20
INTEGER CIT(3,2),CRD(10,2),NFL(10),FL	CARCIT21
LOGICAL ERR,MATCH	CARCIT22
DATA MXWC/0/	CARCIT23
C INITIALIZE.	CARCIT24
CALL BUTLER	CARCIT25
IF(MXWC.EQ.0) MXWC=42+((3990/NCHW)/3)*3	CARCIT26
ERR=.FALSE,	CARCIT27
FLD=BLANKS	CAYSIT28
CALL MOVE(0,NFL,-NFIELD)	CARCIT29
C BEGIN CARD INPUT. STORE CARDS IN CHD ARRAY.	CARCIT30
KF=0	CARCIT31
L=0	CARCIT32
C MAJOR LOOP FOR CARD INPUT.	CARCIT33
IF(.NOT.(CONCAR.OR.BLACAR)) GO TO 23	CARCIT34
10 CALL INCARD	CARCIT35
HOLD=.FALSE,	CARCIT36
IF(BLACAR.AND.L.EQ.0) GO TO 10	CARCIT37
20 CONTINUE	CARCIT38
IF(BLACAR.OR.ENDCAR.OR.CONCAR) GO TO 53	CARCIT39
IF(L.GE.MXCAR) GO TO 904	CARCIT40
L=L+1	CARCIT41
CALL MOVE(CARD,CRD(1,L),NWC)	CARCIT42
CALL CHOP(CARD,0,FLD,0,MXFL)	CARCIT43
IF(.NOT.MATCH(FLD,BLANKS)) GO TO 30	CARCIT44
IF(L.EQ.1) GO TO 901	CARCIT45
GO TO 40	CARCIT46
30 CALL JUST(FLD,0,MXFL,0,0)	CARCIT47
K=LOOK(FIELD,NFIELD,FLD)	CARCIT48
IF(K.GT.NFIELD) GO TO 907	CARCIT49
IF(K.LE.KF) GO TO 903	CARCIT50
IF(KF.GT.0) NFL(KF)=N	CARCIT51
KF=K	CARCIT52
N=0	CARCIT53
49 N=N+1	CARCIT54
GO TO 10	CARCIT55
C END OF CARD INPUT.	CARCIT56
53 NCRD=L	CARCIT57
NFL(K)=N	CARCIT58

IF(NCRD,LE,0) RETURN	CARCIT59
C MOVE CHARACTERS FROM CRD TO CIT.	CARCIT60
JJ=3*NFIELD*NCHW	CARCIT61
LB=0	CARCIT62
DO 80 K=1,NFIELD	CARCIT63
NC=0	CARCIT64
LAST=JJ	CARCIT65
NRT=0	CARCIT66
IF(NFL(K),LE,0) GO TO 70	CARCIT67
LA=LB+1	CARCIT68
LB=LB+NFL(K)	CARCIT69
DO 60 L=LA,LB	CARCIT73
M=0	CARCIT71
N=KARB-KARA+1	CARCIT72
IF(L,EQ,LR) CALL SCOUT(1,0,BLANK,CRD(1,L),KARA-1,N,M)	CARCIT73
N=N-M	CARCIT74
IF(JJ+N,GT,MXWC*NCHW) GO TO 904	CARCIT75
CALL CHOP(CRD(1,L),KARA-1,CIT,JJ,N)	CARCIT76
CALL SPOT(L,N,M,JJ,4HIIII,24HL N JJ)	CARCIT77
60 JJ=JJ+N	CARCIT78
NC=JJ-LAST	CARCIT79
CALL JUST(CIT,LAST,NC,NSUB,NADD)	CARCIT80
JJ=JJ-NSUB	CARCIT81
NC=JJ-LAST	CARCIT82
CALL ARTY(CIT,LAST,NC,NRT)	CARCIT83
JJ=JJ+NADD	CARCIT84
CALL SPOT(NC,LAST,NRT,NADD,4HIIII,24HNC LAST NRT NADD)	CARCIT85
CALL TALMUD(FIELD(K),CIT,LAST,NC)	CARCIT86
70 CIT(1,K)=NC	CARCIT87
CIT(2,K)=LAST	CARCIT88
80 CIT(3,K)=NRT	CARCIT89
C NORMAL RETURN,	CARCITYG
RETURN	CARCIT91
C ERRORS.	CARCIT92
901 KE=1	CARCIT93
GO TO 910	CARCIT94
902 KE=2	CARCIT95
GO TO 910	CARCIT96
903 KE=3	CARCIT97
GO TO 910	CARCIT98
904 KE=4	CARCIT99
910 NCRD=L	CARCIT100
CALL PAGER(-1)	CARCIT101
DO 920 L=1,NCRD	CARCIT102
CALL PAGER(1)	CARCIT103
920 PRINT 9000, (CRD(J,L),J=1,NWC)	CARCIT104
9003 FORMAT(1X,20(1H*),10A8,20(1H*))	CARCIT105
CALL PAGER(2)	CARCIT106
GO TO (931,932,933,934),KE	CARCIT107
931 PRINT 9031	CARCIT108
9031 FORMAT(53H THE FIRST CARD OF A CITATION MUST HAVE A FIELD LABEL)	CARCIT109
GO TO 940	CARCIT110
932 PRINT 9032,FLD	CARCIT111
9032 FORMAT(17H THE FIELD LABEL ,A4,37H IS MISSPELLED OR OTHERWISE INCO	CARCIT112
,RECT)	CARCIT113
GO TO 940	CARCIT114
933 PRINT 9033	CARCIT115

9033	FORMAT(24H FIFLCS 4RE OUT OF ORDER)	CARCI116
	GO TO 943	CARCI117
934	PRINT 9034	CARCI118
9034	FORMAT(20H CITAT13Y IS TOO BIG)	CARCI119
940	PRINT 9040	CARCI120
9040	FORMAT(23H SKIPPING THIS CITATION)	CARCI121
	IF(ENDCAR.OR,BLACAR.OR,CONCAR) GO TO 960	CARCI122
950	CALL INCARD	CARCI123
	IF(ENDCAR.OR,BLACAR.OR,CONCAR) GO TO 960	CARCI124
	CALL PAGER(1)	CARCI125
	PRINT 9000, (CARD(J),J=1,NWC)	CARCI126
	GO TO 950	CARCI127
960	CALL PAGER(-1)	CARCI128
	ERR=.TRUE.	CAPCI129
C	ERROR RETURN,	CAHCI130
	RETURN	CARCI131
C	INTERNAL FORTRAN VARIABLES ARE ,,,	CARCI132
C	FLD	CARCI133
C	IC	CARCI134
C	JJ	CARCI135
C	K	CARCI136
C	KF	CARCI137
C	L	CARCI138
C	LB	CARCI137
C	LA	CAHCI140
C	LAST	CARCI141
C	MXWC	CARCI142
C		CARCI143
C	NC	CARCI144
C	NFL(10)	CARCI145
C	NL	CARCI146
C	NRT	CARCI147
	END	CARCI148

	IDENT CHAR, LIST 3,E,M EJECT	KHAR(G,J)	CHAR(CH,G,J)	CHAR	
				CHAR	2
				CHAR	3
				CHAR	4
				CHAR	5
*	SUBROUTINE CHAR(CH,G,J)			CHAR	6
*	PUTS A CHARACTER INTO A CHARACTER ARRAY.			CHAR	7
*	THE CHARACTER IS OBTAINED FROM CH, IN WHICH IT IS RIGHT JUSTIFIED			CHAR	8
*	WITH BINARY ZERO FILL ON THE LEFT,			CHAR	9
*	IT IS PUT INTO THE J-TH CHARACTER POSITION OF THE ARRAY G, COUNTING			CHAR	10
*	EIGHT CHARACTERS PER WORD AND NUMBERING FROM LEFT TO RIGHT IN A WORD,			CHAR	11
*	OTHER CHARACTERS IN 3 ARE UNCHANGED,			CHAR	12
*	6400 TIME = 14.5 MICROSECONDS (APPROX.)			CHAR	13
*				CHAR	14
*	INTEGER FUNCTION KHAR(G,J)			CHAR	15
*	GETS A CHARACTER FROM A CHARACTER ARRAY.			CHAR	16
*	THE CHARACTER IS OBTAINED FROM THE J-TH CHARACTER POSITION OF ARRAY G,			CHAR	17
*	COUNTING EIGHT CHARACTERS PER WORD AND NUMBERING FROM LEFT TO			CHAR	18
*	RIGHT WITHIN WORDS.			CHAR	19
*	IT IS STORED IN THE FUNCTION RESULT WORD IN THE RIGHT-MOST POSITION			CHAR	20
*	WITH BINARY ZERO FILL TO THE LEFT.			CHAR	21
*	6400 TIME = 10.7 MICROSECONDS (APPROX.)			CHAR	22
*				CHAR	23
*	PROGRAMMED BY ROBERT M. RUSSELL, DECEMBER 1967.			CHAR	24
*				CHAR	25
	ENTRY CHAR			CHAR	26
	VFD 24/4HCHAR,36/3			CHAR	27
CHAR	BSS 1			CHAR	28
	SB1 X1	MODIFICATION TO ALLOW THIS ROUTINE		JRT	29
	SA2 A1+1	TO WORK ON FTN COMPILER		JRT	30
	SB2 X2	INSTEAD OF RUN COMPILER		JRT	31
	SA2 A1+2			JRT	32
	SB3 X2			JRT	33
	SA1 B3	X1=J		CHAR	34
	SX1 X1-1	X1=J-1		CHAR	35
	SX0 7			CHAR	36
	RX2 X1*X0	X2=INDEX OF CHAR WITHIN WORD		CHAR	37
	AX1 3	X1=INDEX OF G WORD		CHAR	38
	SA3 X2+SHIFT	X3=SHIFT		CHAR	39
	SH4 X3	B4=SHIFT		CHAR	40
	SA2 31	X2=CH		CHAR	41
	SA4 X1+B2	X4=G WORD		CHAR	42
	LX4 B4,X4	X4=SHIFTED G WORD		CHAR	43
	MX0 54			CHAR	44
	BX6 X4*X0	X6=MASKED G WORD		CHAR	45
	BX6 X6*X2	X6=G WORD WITH CH INSERTED		CHAR	46
	SB4 B4-60	B4=RETURN SHIFT		CHAR	47
	AX6 B4,X6	X6=RESULT		CHAR	48
	SA6 A4	STORE RESULT		CHAR	49
	JP CHAR			CHAR	50
	ENTRY KHAR			CHAR	51
	VFD 24/4HKHAR,36/2			CHAR	52
KHAR	BSS I			CHAR	53
	SB1 X1	MODIFICATION TO ALLOW THIS ROUTINE		JHT	54
	SA2 A1+1	TO WORK ON FTN COMPILER		JHT	55
	SB2 X2	INSTEAD OF RUN COMPILER		JRT	56
	SA2 A1+2			JRT	57
	SB3 X2			JHT	58
	SA1 B2	X1=J		CHAR	59

	SX1	X1-1	X1=J-1	CHAR	49
	SX0	7		CHAR	50
	BX2	X1*X0	X2=INDEX OF CHAR WITHIN WORD	CHAR	51
	AX1	3	X1=INDEX OF G WORD	CHAR	52
	SA3	X2+SHIFT	X3=SHIFT	CHAR	53
	SB4	X3	B4=SHIFT	CHAR	54
	SA1	X1+B1	X1=WORD	CHAR	55
	LX6	B4,X1	X6=SHIFTED WORD	CHAR	56
	SX0	778		CHAR	57
	BX6	X6*X0	X6=RESULT	CHAR	59
	JP	KHAR		CHAR	59
SHIFT	DATA	6		CHAR	60
	DATA	12		CHAR	61
	DATA	18		CHAR	62
	DATA	24		CHAR	63
	DATA	30		CHAR	64
	DATA	36		CHAR	65
	DATA	42		CHAR	66
	DATA	48		CHAR	67
	END			CHAR	68

SUBROUTINE CHOP(X,LX,Y,LY,NC)	CHOP	2
C MOVES A STRING OF NC CONSECUTIVE CHARACTERS (PACKED NCHW CHARACTERS	CHCF	3
C PER CELL) FROM ARRAY X TO ARRAY Y, STARTING IN X AT CHARACTER NUMBER	CHOF	4
C LX+1 AND IN Y AT CHARACTER NUMBER LY+1.	CHOP	5
C IF NC IS ZERO, NO ACTION OCCURS.	CHOP	6
C IF NC IS NEGATIVE, ONLY CHARACTER LX+1 IN X IS MOVED TO Y AND IT IS	CHOP	7
C MOVED INTO EACH OF THE -NC POSITIONS OF Y, STARTING AT LY+1.	CHOP	8
C X AND Y MAY OVERLAP IN STORAGE PROVIDED THE Y CHARACTER AT LY+1 COMES	CHOP	9
C BEFORE THE X CHARACTER AT LX+1 IN STORAGE.	CHOP	10
C FOR SPEED, IF IS ADVISABLE TO HAND CODE THIS SUBROUTINE.	CHOP	11
C PROGRAMMED BY ROBERT M. RUSSELL, JANUARY, 1969.	CHUP	12
INTEGER X(2),Y(2)	CHOP	13
IF(IABS(NC).EQ.0) RETURN	CHOP	14
KX=LX	CHOP	15
KY=LY	CHOP	16
IF(NC.LT.0) GO TO 20	CHOP	17
DO 10 J=1,NC	CHOP	18
KH=KHAR(X,KX+J)	CHOP	19
10 CALL CHAR(KH,Y,KY+J)	CHOP	20
RETURN	CHOP	21
20 NNC=-NC	CHOP	22
KH=KHAR(X,KX+1)	CHOP	23
DO 30 J=1,NNC	CHOP	24
30 CALL CHAR(KH,Y,KY+J)	CHOP	25
RETURN	CHOP	26
END	CHOP	27

SUBROUTINE CIPHER(INT,LINE,LL,NL)	CIPHER 2
C CONVERTS AN INTEGER, INT, INTO A STRING OF DIGIT CHARACTERS AND PUTS	CIPHER 3
C THEM IN THE CHARACTER ARRAY LINE PRECEDED WITH A COMMA AND A BLANK.	CIPHER 4
C THESE CHARACTERS AUGMENT CHARACTERS ALREADY IN LINE UP TO ANG	CIPHER 5
C INCLUDING THE LL-TH POSITION, NL (GIVEN) IS THE NUMBER OF POSITIONS	CIPHER 6
C STILL AVAILABLE IN LINE.	CIPHER 7
C LL IS THEN INCREMENTED TO COVER THE ADDED CHARACTERS, AND NL IS	CIPHER 8
C DECREMENTED THE SAME AMOUNT,	CIPHER 9
C THERE IS ONE ABNORMAL CASE, WHEN INT IS TOO LARGE TO FIT IN THE NL	CIPHER10
C AVAILABLE CHARACTER POSITIONS IN LINE, NAMELY WHEN INT HAS MORE THAN	CIPHER11
C NL-3 DIGITS. IN THIS CASE, THE COMMA IS PUT IN LINE, THE REMAINING	CIPHER12
C POSITIONS ARE FILLED WITH BLANKS, AND NL IS SET TO 0 BEFORE EXITING.	CIPHER13
C NL-3 IS THE MAXIMUM NUMBER OF DIGITS ALLOWED IN INT IN ORDER TO	CIPHER14
C PROVIDE SPACE FOR A TRAILING COMMA LATER,	CIPHER15
C INT MUST BE POSITIVE AND TYPE INTEGER,	CIPHER16
INTEGER LINE(1),DIGIT(10)	CIPHER17
DATA (DIGIT(1),1=1,10)/1H0,1H1,1H2,1H3,1H4,1H5,1H6,1H7,1H8,1H9/	CIPHER18
CALL SPOT(INT,LL,NL,0,4HIIII,24HINT LL NL)	CIPHER19
IF(LL.LE.0) GO TO 5	CIPHER20
C INSERT COMMA AND BLANK IN LINE.	CIPHER21
N=MIN0(2,NL)	CIPHER22
CALL CHOP(2H, ,0,LINE,LL,N)	CIPHER23
LL=LL+N	CIPHER24
NL=NL-N	CIPHER25
C CALCULATE THE NUMBER OF DIGITS (NUMDIG) IN INT,	CIPHER26
S IQ=INT	CIPHER27
M=10	CIPHER28
DO 10 NUMDIG=1,100	CIPHER29
IF(IQ.LT,M) GO TO 20	CIPHER30
10 M=M*10	CIPHER31
20 IF(NUMDIG.LT,NL) GO TO 30	CIPHER32
C NOT ENOUGH ROOM IN LINE. FILL WITH BLANKS AND RETURN.	CIPHER33
CALL CHOP(1H ,0,LINE,LL,-NL)	CIPHER34
NUMDIG=NL	CIPHER35
GO TO 50	CIPHER36
C CONVERT INT, PUT DIGIT CHARACTERS IN LINE,	CIPHER37
30 DO 40 J=1,NUMDIG	CIPHER38
IR=1+MOD(IQ,10)	CIPHER39
CALL CHAR(KHAR(DIGIT(IR),1),LINE,LL+NUMDIG-J+1)	CIPHER40
40 IQ=IQ/10	CIPHER41
C UPDATE LL AND NL.	CIPHER42
50 LL=LL+NUMDIG	CIPHER43
NL=NL-NUMDIG	CIPHER44
C CALL SPOT(NUMDIG,LL,NL,0,4HIIII,24HNUMDIGLL NL	CIPHER45
RETURN	CIPHER46
END	CIPHER47


```

SUBROUTINE COPCIT(CITX,CITY,NW)
C COPIES A CITATION LOCATED IN ARRAY CITX INTO THE ARRAY CITY,
C SETS NW = THE NUMBER OF CELLS (MACHINE WORDS) IN THE CITATION.
COMMON /CORE/NCHW,BLANKS,BLANK
INTEGER BLANKS,BLANK
COMMON /FIEL/FIELD(10),IFIELD(10),NFIELD,KDESCR
INTEGER FIELD
INTEGER CITX(3,2),CITY(3,2)
DATA MXWC/540/
CALL BUTLER
NC=CITX(1,NFIELD)
LC=CITX(2,NFIELD)
NN=LC+NC
IF(MIN0(NC,LC).LT.0) CALL HALT
IF(MAX0(NC,LC,NN).GT.MXWC*NCHW) CALL HALT
NW=(NN+NCHW-1)/NCHW
CALL MOVE(CITX,CITY,NW)
RETURN
END

```

```

COPCIT 2
COPCIT 3
COPCIT 4
COPCIT 5
COPCIT 6
COPCIT 7
COPCIT 8
COPCIT 9
COPCIT10
COPCIT11
COPCIT12
COPCIT13
COPCIT14
COPCIT15
COPCIT16
COPCIT17
COPCIT18
COPCIT19
COPCIT20

```

IDENT	DATE,	DATE(D)	DATE	
OBJECT			DATE	2
* DATE(D)			DATE	3
* OBTAINS THE CURRENT DATE FROM THE SCOPE CPC ROUTINE AND STORES IT IN			DATE	4
* THE WORD D,			DATE	5
* FOR EXAMPLE, ON FEBRUARY 29, 1968, D WOULD BE RETURNED AS			DATE	6
* 02/29/68			DATE	7
* THIS IS EIGHT CHARACTERS FOLLOWED BY TWO BLANKS, SO IT CAN BE PRINTED			DATE	8
* IN Ad FORMAT.			DATE	9
* REFERENCE... SEE DATE FUNCTION IN SECTION 3.5.4 (SYSTEM ACTION			DATE	10
* REQUESTS) OF THE SCOPE REFERENCE MANUAL, PAGE 3-38 OF THE APRIL 1967			DATE	11
* EDITION.			DATE	12
* PROGRAMMED BY ROBERT M. RUSSELL, FEBRUARY 1968.			DATE	13
EXT CPC			DATE	14
ENTRY DATE			DATE	15
+ VFD 24/4HDATE,36/1			DATE	16
DATE ASS 1			DATE	17
SX6 B0	X6=0		DATE	18
SA6 STATUS	STORE 0 IN STATUS		DATE	19
SX6 B1	X6=B1=LCC OF D		DATE	20
+ SA6 SAVEB1	STORE X6=B1 IN SAVEB1		DATE	21
RJ CPC	CALL CPC		DATE	22
+ VFD 18/3HTIM,2/3,4/0,12/1,6/0,18/STATUS			DATE	23
SA1 STATUS	X1=DATE AS TYPED BY OPERATOR		DATE	24
EX6 X1	X6=X1		DATE	25
SX2 556	X2=LOW ORDER BLANK CHAR		DATE	26
SX3 778	X3=LOW ORDER CHAR MASK		DATE	27
SB3 10	B3=LOOP COUNTER		DATE	28
LOOP LX6 6	ROTATE X6 ONE CHAR LEFT		DATE	29
BX5 X6*X3	"ASK OFF HIGH ORDER 9 CHARS		DATE	30
BX5 X5-X2	EXCL OR WITH BLANK		DATE	31
NZ X5, FIX	IF NZ, CHAR IS NONBLANK		DATE	32
SB3 B3-1	DECREMENT B3 COUNTER		DATE	33
PL B3, LOOP			DATE	34
FIX LX6 54	UNROTATE X6		DATE	35
SA2 SAVEB1	X2=LCC OF D		DATE	36
SA0 X2	STORE X6 IN 3		DATE	37
J? DATE			DATE	38
SAVEB1 BSS 1			DATE	39
STATUS BSS 1			DATE	40
END			DATE	41
			DATE	42

SUBROUTINE DIVEST	DIVEST 2
C PRINTS THE CURRENT INPUT CARD IF IT IS BEING HELD FOR PRINTING.	DIVEST 3
COMMON /INPT/CARD(10),KOLUM,HOLD,ENDCAR,BLACAR,CONCAR,ERRIN,NWC,	DIVEST 4
, MXFL,MAXCAR,NCARIN,KHR,KIN,LIT,LITX,SYM(5),NUMB,NCSYM,CATLOG,	DIVEST 5
, KARA,KARB	DIVEST 6
INTEGER CARD,SYM	DIVEST 7
LOGICAL HOLD,ENDCAR,BLACAR,CONCAR,ERRIN,CATLOG	DIVEST 8
IF(.NOT.HOLD) RETURN	DIVEST 9
HOLD=.FALSE,	DIVEST13
CALL PAGER(1)	DIVEST11
PRINT 1000, (CARD(J),J=1,NWC)	DIVEST12
1000 FORMAT(14H INPUT CARD...,10A8)	DIVEST13
RETURN	DIVEST14
END	DIVEST15

SUBROUTINE FABRIC(KONCAR)	FABRIC 2
C COLLECTS SYMBOLS FORMING A STATEMENT NAME AND RETURNS AN INDICATOR,	FABRIC 3
C KONCAR, SHOWING WHICH STATEMENT IT IS,	FABRIC 4
C TURNS OFF THE CONCAR INDICATOR.	FABRIC 5
COMMON /CORE/NCHW,BLANKS,BLANK	FABRIC 6
INTEGER BLANKS,BLANK	FABRIC 7
COMMON /INPT/CARD(10),KOLUM,HOLD,ENDCAR,BLACAR,CONCAR,ERRIN,NWC,	FABRIC 8
MXFL,MAXCAR,NCARIN,KHR,KIN,LIT,LITX,SYN(5),NUMB,NCSYM,CATLOG,	FABRIC 9
KARA,KARB	FABRIC10
INTEGER CARD,SYM	FABRIC11
LOGICAL HOLD,ENDCAR,BLACAR,CONCAR,ERRIN,CATLOG	FABRIC12
INTEGER S(3,21)	07/18/71
DATA S(1,1),S(2,1),S(3,1)/8HID	/ FABRIC14
DATA S(1,2),S(2,2),S(3,2)/8HFIELDS	/ FABRIC15
DATA S(1,3),S(2,3),S(3,3)/8HSELECT	/ FABRIC16
DATA S(1,4),S(2,4),S(3,4)/8HNEW ID	/ FABRIC17
DATA S(1,5),S(2,5),S(3,5)/8HPRINT	/ FABRIC18
DATA S(1,6),S(2,6),S(3,6)/8HCITATION	/ FABRIC19
DATA S(1,7),S(2,7),S(3,7)/8HDELETE	/ FABRIC20
DATA S(1,8),S(2,8),S(3,8)/8HREPLACE	/ FABRIC21
DATA S(1,9),S(2,9),S(3,9)/8HORIGINAL	/ FABRIC22
DATA S(1,10),S(2,10),S(3,10)/8HSEARCH	/ FABRIC23
DATA S(1,11),S(2,11),S(3,11)/8HVOCABULA	/ FABRIC24
DATA S(1,12),S(2,12),S(3,12)/8HDEBUG	/ FABRIC25
DATA S(1,13),S(2,13),S(3,13)/8HDESCRIPT	/ FABRIC26
DATA S(1,14),S(2,14),S(3,14)/8HWRITE TA	/ FABRIC27
DATA S(1,15),S(2,15),S(3,15)/8HWIDTH	/ FABRIC28
DATA S(1,16),S(2,16),S(3,16)/8HCATALOG	/ FABRIC29
DATA S(1,17),S(2,17),S(3,17)/8HPUNCH	/ FABRIC30
DATA S(1,18),S(2,18),S(3,18)/8HNUMBERS	/ FABRIC31
DATA S(1,19),S(2,19),S(3,19)/8HPRINT BY	/ FABRIC32
DATA S(1,20),S(2,20),S(3,20)/8HFIELDS	/ 07/18/72
DATA S(1,21),S(2,21),S(3,21)/8HNO NUMBE	/ 07/18/73
CALL BUTLER	FABRIC33
NJ=3	FABRIC34
NK=21	07/18/74
KONCAR=0	FABRIC36
MAXC=NJ*NCHW	FABRIC37
KE=1	FABRIC38
IF(.NOT.CONCAR) GO TO 900	FABRIC39
CALL SYMBOL	FABRIC40
KE=2	FABRIC41
IF(LIT.NE.3) GO TO 900	FABRIC42
CALL SYMBOL	FABRIC43
KE=3	FABRIC44
IF(LIT.NE.1) GO TO 900	FABRIC45
KE=4	FABRIC46
IF(NCSYM.GT.MAXC) GO TO 900	FABRIC47
LO 10 K=1,NK	FABRIC48
CALL SIMILE(1,SYM,0,MAXC,S(1,K),0,MAXC,KISS)	FABRIC49
IF(KISS.EQ.1) GO TO 20	FABRIC50
10 CONTINUE	FABRIC51
KE=5	FABRIC52
GO TO 900	FABRIC53
20 KONCAR=K	FABRIC54
CALL SYMBOL	FABRIC55
	FABRIC56

KE=6	FABRIC57
IF(LIT.NE.3) GO TO 900	FABRIC58
CALL TALMUD(6HFABRIC,S(1,KONCAR),0,3*NCHW)	FABRIC59
CALL KORAN(6HKONCAR,KONCAR,1)	FABRIC60
RETURN	FABRIC61
C ERROR ANALYSIS,	FABRIC62
900 CALL PAGER(2)	FABRIC63
CALL DIVEST	FABRIC64
GO TO (901,902,902,903,903,902),KE	FABRIC65
C NOT A STATEMENT CARD,	FABRIC66
901 PRINT 9001	FABRIC67
9001 FORMAT(46H THIS CARD DOES NOT START A STATEMENT PROPERLY, 82(1H*))	FABRIC68
GO TO 910	FABRIC69
C INCORRECT STATEMENT CARD,	FABRIC70
902 PRINT 9002	FABRIC71
9002 FORMAT(25H INCORRECT STATEMENT CARD,103(1H*))	FABRIC72
GO TO 910	FABRIC73
C INCORRECT STATEMENT NAME,	FABRIC74
903 PRINT 9003	FABRIC75
9003 FORMAT(25H INCORRECT STATEMENT NAME,103(1H*))	FABRIC76
C FINALIZE ERROR ANALYSIS,	FABRIC77
910 ERRIN=.TRUE,	FABRIC78
PRINT 9050, KE	FABRIC79
9050 FORMAT(118X,6HFABRIC I4)	FABRIC80
KONCAR=0	FABRIC81
CALL PAGER(-1)	FABRIC82
RETURN	FABRIC83
END	FABRIC84

IDENT	FERRET,	FERRET(SUBA,LA,SUBB,LB)	FERRET 2
LIST	G,E,M		FERRET 3
EJECT			FERRET 4
* LOCATES SUBROUTINE NAME AND LOC OF CALL, FOR THIRD AND FOURTH CALLS UP			FERRET 5
* THE SUBROUTINE CALL TREE, NAMELY,,,			FERRET 6
* SUBROUTINE BB			FERRET 7
9 CALL AA			FERRET 8
*			FERRET 9
* SUBROUTINE AA			FERRET10
* CALL BUTLER			FERRET11
9			FERRET12
9 SUBROUTINE BUTLER			FERRET13
* CALL FERRET(SUBA,LA,SUBB,LB)			FERRET14
*			FERRET15
* THE PURPOSE OF FERRET IS SO BUTLER, A DEBUGGING SUBROUTINE, CAN PRINT			FERRET16
* THE NAME OF A SUBROUTINE AND LOCATION IN IT WHERE YOU ARE CALLING			FERRET17
* ANOTHER SUBROUTINE,			FERRET18
* PROGRAMMED BY ROBERT M. RUSSELL, JANUARY, 1968.			FERRET19
ENTRY FERRET			FERRET20
VFD 36/6HFERRET,24/0			FERRET21
FERRET BSS 1			FERRET22
SB1 X1	MODIFICATION TO ALLOW THIS ROUTINE	JRT 36	
SA2 A1+1	TO WORK ON FTN COMPILER	JRT 57	
Sd2 X2	INSTEAD OF RUN COMPILER	JRT 58	
SA2 A1+2	MODIFICATION	JRT 39	
SB3 X2	MODIFICATION	JRT 40	
SA2 A1+3	MODIFICATION	JRT 41	
SB4 X2	MODIFICATION	JRT 42	
MX0 18			FERRET23
LX0 18	X0=MASK FOR LOW-ORDER 18 BITS		FERRET24
SA1 FERRET			FERRET25
LX1 30			FERRET26
BX1 X1*X0			FERRET27
SX1 X1-1	X1=LOC IN BUTLER OF RJ FERRET		FERRET28
SA1 X1			FERRET29
BX1 X1*X0			FERRET30
SX1 X1+1	X1=LOC OF BUTLER BSS 1		FERRET31
SA1 X1			FERRET32
LX1 30			FERRET33
BX1 X1*X0			FERRET34
SX1 X1-1	X1=LOC IN PA OF RJ BUTLER		FERRET35
BX6 X1	SAVE, LOC IN AA OF RJ BUTLER		FERRET36
SA1 X1			FERRET37
BX1 X1*X0	X1=LOC IN AA OF NAME AA		FERRET38
IX6 X6-X1			FERRET39
SA6 82	STORE* REL LOC IN AA OF RJ BUTLER		FERRET40
SA2 X1	X2=NAME OF AA		FERRET41
BX6 X2			FERRET42
SA6 B1	STORE NAME OF AA		FERRET43
SX1 X1+1	X1=LOC IN AA OF AA BSS 1		FERRET44
SA1 X1			FERRET45
LX1 30			FERRET46
BX1 X1*X0			FERRET47
SX1 X1-1	X1=LOC IN BB OF RJ AA		FERRET48
BX6 X1	SAVE, LOC IN Yd OF RJ AA		FERRET49
SA1 X1			FERRET50
BX1 X1*X0	X1=LOC IN BB OF NAME BB		FERRET51

IX6 X6-X1
SA6 B4
SA1 X1
BX6 X1
SA6 B3
JP FERRET
END

STORE, REL LOC IN B3 OF RJ 44
X1=NAME OF BR

STORE NAME OF BB

FERRET52
FERRET53
FERRET54
FERRET55
FERRET56
FERRET57
FERRET58

SUBROUTINE GETCH	GETCH 2
C OBTAINS THE NEXT CHARACTER FROM THE INPUT CARDS,	GETCH 3
C STORING ITS CODE VALUE IN KH AND ITS TYPE	GETCH 4
C IN KIN, IN INPT COMMON.	GETCH 5
COMMON /CORE/NCHW,BLANKS,BLANK	GETCH 6
INTEGER BLANKS,BLANK	GETCH 7
COMMON /INPT/CARD(10),KOLUM,HOLD,ENDCAR,BLACAR,CONCAR,ERRIN,NWC,	GETCH 8
MXFL,MAXCAR,NCARIN,KHR,KIN,LIT,LITX,SYM(5),NUMB,NCSYM,CATLOG,	GETCH 9
, KARA,KARB	GETCH 13
INTEGER CARD,SYM	GETCH 11
LOGICAL HOLD,ENDCAR,BLACAR,CONCAR,ERRIN,CATLOG	GETCH 12
IF(KOLUM,LT,80) GO TO 10	GETCH 13
CALL INCARD	GETCH 14
IF(.NOT.ENDCAR) GO TO 10	GETCH 15
KHR=BLANK	GETCH 16
KIN=13	GETCH 17
RETURN	GETCH 18
13 KOLUM=KOLUM+1	GETCH 19
KHR=KHAR(CARD,KOLUM)	GETCH 20
KIN=IDEAL(KHR)	GETCH 21
RETURN	GETCH 22
END	GETCH 23

SUBROUTINE GETLAB(LAB,MIN,MAX,N,KERR)	GETLAB 2
C COLLECTS FROM INPUT CARDS, SYMBOLS FORMING A LIST OF FIELD LABELS AND	GETLAB 3
C STORES THEM IN MACHINE-USABLE FORM,	GETLAB 4
C THE LABELS ARE STORED IN THE ARRAY LAB, ONE LABEL PER CELL, AND THEIR	GETLAB 5
C NUMBER IS STORED IN N.	GETLAB 6
C MAX (GIVEN) IS THE MAXIMUM LENGTH OF LAB,	GETLAB 7
C MIN (GIVEN) IS THE MINIMUM LENGTH OF LAB,	GETLAB 8
C SETS KERR=0 IF NO ERRORS DETECTED, KERR=1 IF N IS LESS THAN MIN OR	GETLAB 9
C GREATER THAN MAX, AND KERR=2 IF A SYNTAX ERROR OCCURRED, A MESSAGE IS	GETLAB10
C PRINTED FOR AN ERROR.	GETLAB11
C WHEN N EXCEEDS MAX, ONLY MAX LABELS ARE STORED IN LAB, BUT SYMBOL	GETLAB12
C COLLECTION CONTINUES UNTIL THE END OF THE LIST (UNLESS A SYNTAX ERROR	GETLAB13
C OCCURS FIRST) AND THE ACTUAL N IS RETURNED.	GETLAB14
C USES SUBROUTINE SYMBOL FOR THE ACTUAL COLLECTION OF SYMBOLS,	GETLAB15
C ASSUMES, AT ENTRY, THAT THE LEFT PARENTHESIS IS THE NEXT SYMBOL TO	GETLAB16
C APPEAR, AT EXIT, THE RIGHT PARENTHESIS IS THE LAST SYMBOL THAT WAS	GETLAB17
C COLLECTED.	GETLAB18
COMMON /CORE /NCHW,BLANKS,BLANK	GETLAB19
INTEGER BLANKS,BLANK	GETLAB20
COMMON /INPT/CARD(10),KOLUM,HOLD,ENDCAR,BLACAR,CONCAR,ERRIN,NWC,	GETLAB21
MXFL,MAXCAR,NCARIN,KHR,KIN,LIT,LITX,SYM(5),NUMB,NCSYM,CATLOG,	GETLAB22
KARA,KARB	GETLAB23
INTEGER CARD,SYM	GETLAB24
LOGICAL HOLD,ENDCAR,BLACAR,CONCAR,ERRIN,CATLOG	GETLAB25
INTEGER LAB(2)	GETLAB26
KERR=0	GETLAB27
K=0	GETLAB28
CALL SYMBOL	GETLAB29
IF(LIT,NE.4) GO TO 901	GETLAB30
10 CALL SYMBOL	GETLAB31
IF(LIT,NE.1) GO TO 901	GETLAB32
IF(NCSYM,GT,MXFL) GO TO 902	GETLAB33
K=K+1	GETLAB34
IF(K,LE,MAX) LAB(K)=SYM(1)	GETLAB35
CALL SYMBOL	GETLAB36
IF(LIT,EQ.5) GO TO 20	GETLAB37
IF(LIT,NE.6) GO TO 901	GETLAB38
GO TO 10	GETLAB39
20 CALL SCOUT(0,0,BLANK,CARD,0,KOLUM-1,NF)	GETLAB40
IF(NF,LT,KOLUM-1) CALL DIVEST	GETLAB41
N=K	GETLAB42
IF(K,GE,MIN,AND,K,LT,MAX) GO TO 100	GETLAB43
IF(K,LT,MIN) KE=3	GETLAB44
IF(K,GT,MAX) KE=4	GETLAB45
KERR=1	GETLAB46
GO TO 910	GETLAB47
100 CONTINUE	GETLAB48
CALL SPOT(MIN,MAX,N,KERR,4HIIII,24HMIN MAX N KERR)	GETLAB49
RETURN	GETLAB50
901 KE=1	GETLAB51
KERR=2	GETLAB52
GO TO 910	GETLAB53
902 KE=2	GETLAB54
KERR=2	GETLAB55
GO TO 910	GETLAB56
910 IF(.NOT.(KOLUM,EQ.1.AND,KIN,EQ.4)) CALL DIVEST	GETLAB57
CALL PAGER(2)	GETLAB58

GO TO (921,922,923,924),KE	GETLAB59
C SYNTAX ERROR,	GETLAB60
921 PRINT 9001	GETLAB61
9001 FORMAT(37H SYNTAX ERROR IN LIST OF FIELD LABELS,5X,86(1H*))	GETLAB62
GO TO 930	GETLAB63
C MISPELLED FIELD LABEL,	GETLAB64
922 PRINT 9002, SYM	GETLAB65
9002 FORMAT(34H THIS FIELD LABEL IS MISPELLED),...,5A8,5X,49(1H*))	GETLAB66
GO TO 930	GETLAB67
C N IS LESS THAN MIN,	GETLAB68
923 PRINT 9003, K	GETLAB69
9003 FORMAT(15H THERE ARE ONLY,13,13H FIELD LABELS,5X,92(1H*))	GETLAB70
GO TO 930	GETLAB71
C N IS GREATER THAN MAX,	GETLAB72
924 PRINT 9004, K,MAX	GETLAB73
9004 FORMAT(10H THERE ARE,13,30H FIELD LABELS. THE MAXIMUM IS,13,5X,	GETLAB74
, 77(1H*))	GETLAB75
C FINALIZE ERROR ANALYSIS,	GETLAB76
F30 ERRIN=.TRUE.	GETLAB77
PHINT 9050, KE	GETLAB78
9050 FORMAT(118X,6HGETLAB I4)	GETLAB79
CALL PAGER(-1)	GETLAB80
GO TO 100	GETLAB81
END	GETLAB82

	SUBROUTINE GETNUM(NUM,MIN,MAX,N,KERR)	GETNUM 2
C	COLLECTS FROM INPUT, SYMBOLS FORMING A LIST OF CITATION NUMBERS AND	GETNUM 3
C	STORES THEM IN MACHINE-USABLE FORM.	GETNUM 4
C	THE LIST MAY INCLUDE INDICATIONS OF CONSECUTIVE NUMBERS. FOR EXAMPLE,	GETNUM 5
C	(6,13-29,11) MIGHT BE PUNCHED ON THE INPUT CARD. NOTE ALSO THAT THE	GETNUM 6
C	NUMBERS NEED NOT BE IN ASCENDING ORDER.	GETNUM 7
C	THE NUMBERS ARE COLLECTED, CONVERTED TO INTERNAL FORM AND STORED IN	GETNUM 8
C	THE ARRAY NUM, FOLLOWED BY A 0. FOR OUR EXAMPLE, NUM WOULD CONTAIN 6,	GETNUM 9
C	13, -29, 11, 0 ON RETURN. THE NUMBER OF NUMBERS IS STORED IN N, N=4 IN	GETNUM10
C	THE EXAMPLE.	GETNUM11
C	MIN (GIVEN) IS THE MINIMUM LENGTH OF NUM.	GETNUM12
C	MAX (GIVEN) IS THE MAXIMUM LENGTH OF NUM.	GETNUM13
C	SETS KERR=0 IF NO ERRORS DETECTED, KERR=1 IF N IS LESS THAN MIN OR	GETNUM14
C	GREATER THAN, AND KERR=2 IF A SYNTAX ERROR OCCURRED, A MESSAGE IS	GETNUM15
C	PRINTED FOR AN ERROR.	GETNUM16
C	WHEN N EXCEEDS MAX, ONLY MAX NUMBERS ARE STORED IN NUM BUT SYMBOL	GETNUM17
C	COLLECTION CONTINUES UNTIL THE END OF THE LIST AND THE ACTUAL N IS	GETNUM18
C	RETURNED-UNLESS A SYNTAX ERROR OCCURS FIRST.	GETNUM19
C	USES SUBROUTINE SYMBOL FOR THE ACTUAL COLLECTION OF SYMBOLS.	GETNUM20
C	ASSUMES, AT ENTRY, THAT THE LEFT PARENTHESIS IS THE NEXT SYMBOL TO BE	GETNUM21
C	COLLECTED. AT EXIT, THE RIGHT PARENTHESIS IS THE LAST ONE THAT WAS	GETNUM22
C	COLLECTED.	GETNUM23
	COMMON /CORE/NCHW,BLANKS,BLANK	GETNUM24
	INTEGER BLANKS,BLANK	GETNUM25
	COMMON /INPT/CARD(10),KOLUM,HOLD,ENDCAR,BLACAR,CONCAR,ERRIN,NWC,	GETNUM26
	, MXFL,MAXCAR,NCARIN,KHR,KIN,LIT,LITX,SYM(5),NUMB,NCSYM,CATLOG,	GETNUM27
	, KARA,KARB	GETNUM28
	INTEGER CARD,SYM	GETNUM29
	LOGICAL HOLD,ENDCAR,BLACAR,CONCAR,ERRIN,CATLOG	GETNUM30
	INTEGER NUM(2)	GETNUM31
	KERR=0	GETNUM32
	K=0	GETNUM33
	CALL SYMROL	GETNUM34
	IF(LIT.NE.4) GO TO 902	GETNUM35
10	CALL SYMBOL	GETNUM36
	IF(LIT.NE.2) GO TO 902	GETNUM37
	K=K+1	GETNUM38
	N=NUMB	GETNUM39
	IF(LITX.EQ.7) N=-N	GETNUM40
	IF(K.LE.MAX) NUM(K)=N	GETNUM41
	CALL SYMROL	GETNUM42
	IF(LIT.EQ.3) GO TO 20	GETNUM43
	IF(LIT.EQ.7.AND.N.GT.0) GO TO 10	GETNUM44
	IF(LIT.EQ.6) GO TO 10	GETNUM45
	GO TO 902	GETNUM46
20	N=K	GETNUM47
	IF(K.LT.MIN.OR.K.GT.MAX) GO TO 901	GETNUM48
	IF(K.LT.MAX) NUM(K+1)=0	GETNUM49
	CALL SCOUT(0,0,BLANK,CARD,0,KOLUM-1,NF)	GETNUM50
	IF(NF.LT.KOLUM-1) CALL DIVEST	GETNUM51
100	CONTINUE	GETNUM52
	CALL SPOT(MIN,MAX,N,KERR,4HIIII,24HMIN MAX N KERR)	GETNUM53
	RETURN	GETNUM54
901	KERR=1	GETNUM55
	GO TO 910	GETNUM56
902	KERR=2	GETNUM57
910	IF(.NOT.(KOLUM.EQ.1.AND.KIN.EQ.4)) CALL DIVEST	GETNUM58

```

      CALL PAGER(2)
      GO TO (921,922), KERR
C WRONG NUMBER OF ITEMS IN LIST,
  921 PRINT 9001
  9001 FORMAT(30H WRONG NUMBER OF ITEMS IN LIST,5X,93(1H*))
      GO TO 930
C SYNTAX ERROR,
  922 PRINT 9002
  9002 FORMAT(32H SYNTAX ERROR IN LIST OF NUMBERS,5X,91(1H*))
  930 ERRIN=,TRUE,
      PRINT 9050, KERR
  9050 FORMAT(118X,6HGETNUM I4)
      CALL PAGER(-1)
      RETURN
      END

```

```

GETNUM59
GETNUM60
GETNUM61
GETNUM62
GETNUM63
GETNUM64
GETNUM65
GETNUM66
GETNUM67
GETNUM68
GETNUM69
GETNUM70
GETNUM71
GETNUM72
GETNUM73

```

SUBROUTINE GOSSIP(LINGO)	GOSSIP 2
C READS A FREE-FORM STATEMENT FROM INPUT CARDS, FOR EXAMPLE, THE TEXT OF	GOSSIP 3
C AN ID STATEMENT,	GOSSIP 4
C THE TEXT IS STORED IN THE ARRAY LINGO,	GOSSIP 5
COMMON /CORE/NCHW,BLANKS,BLANK	GOSSIP 6
INTEGER BLANKS,BLANK	GOSSIP 7
COMMON /FAM/SLASH,STAR,COMMA,ZERO,MXCHAR,MXCDER,MXWDES,MXCHW,	GOSSIP 8
MXFREE,MXWFRE	GOSSIP 9
INTEGER SLASH,STAR,COMMA,ZERO	GOSSIP10
COMMON /INPT/CARD(10),KOLUM,HOLD,ENDCAR,BLACAR,CONCAR,ERRIN,NWC,	GOSSIP11
MXFL,MAXCAR,NCARIN,KHR,KIN,LIT,LITX,SYM(5),NUMB,NCSYM,CATLOG,	GOSSIP12
KARA,KARB	GOSSIP13
INTEGER CARD,SYM	GOSSIP14
LOGICAL HOLD,ENDCAR,BLACAR,CONCAR,ERRIN,CATLOG	GOSSIP15
CALL BUTLER	GOSSIP16
CALL CHOP(BLANKS,0,LINGO,0,-MXFREE)	GOSSIP17
K=0	GOSSIP18
10 K=K+1	GOSSIP19
IF(K,LE,MXFREE) CALL CHAR(KHR,LINGO,K)	GOSSIP20
CALL GETCH	GOSSIP21
IF(.NOT.(KOLUM,EQ,1.AND,KHR,EQ,SLASH,OR,ENDCAR)) GO TO 10	GOSSIP22
CALL TALMUD(6HGOSSIP,LINGO,0,MXFREE)	GOSSIP23
RETURN	GOSSIP24
END	GOSSIP25

SUBROUTINE HALT	HALT	2
C THIS SUBROUTINE IS CALLED FOR AN UNEXPLAINABLE ERROR CONDITION,	HALT	3
C USUALLY IN DEBUGGING. IT PRINTS TRACE INFORMATION SO YOU CAN TELL	HALT	4
C WHERE IT WAS CALLED, AND THEN STOPS.	HALT	5
COMMON /TEST/ITEST	HALT	6
ITEST=1	HALT	7
CALL BUTLER	HALT	8
STOP	HALT	9
END	HALT	10

	IDENT	HASH.	HASH(G,NG,NTAB,KODE)	HASH	2
*	HASH	(G,NG,NTAB,KODE)		HASH	4
*	COMPUTES	A HASH CODE FOR AN ALPHAMERIC STRING G WHICH IS NG COMPUTER		HASH	5
*	WORDS	IN LENGTH,		HASH	6
*	(SEE	W.D. MAURER, COMM. ACM, JAN, 1968, PAGE 35.)		HASH	7
*	NTAB	IS THE LENGTH OF THE HASH TABLE,		HASH	8
*	KODE	IS THE RESULTING HASH CODE,		HASH	9
*	THE	METHOD IS THE DIVISOR METHOD OF MAURER,		HASH	10
a	TIME	(6400)=28.3+4.0*NG MICROSECONDS (APPROX.)		HASH	11
	ENTRY	HASH		HASH	12
	VFD	24/4HHASH,36/4		HASH	13
HASH	BSS	1		HASH	14
	SB1	X1	MODIFICATION TO ALLOW THIS ROUTINE	JHT	43
	SA2	A1+1	TO WORK ON FTM COMPILER	JRT	44
	S92	X2	INSTEAD OF RUN COMPILER	JRT	45
	SA2	A1+2	MODIFICATION	JRT	46
	S63	X2	MODIFICATION	JRT	47
	SA2	A1+3	MODIFICATION	JRT	48
	SB4	X2	MODIFICATION	JRT	49
	SB6	1	B6=1=CONSTANT	HASH	15
	SA2	B2	X2=NG	HASH	16
	Sd2	X2	B2=NG	HASH	17
	SA1	B1	X1=G(1)	HASH	18
	bX3	X1	X3=G(1)	HASH	19
NEXTW	SB2	B2-B6	COUNT THE G WORDS	HASH	20
	ZR	B2,COSY	EXIT WHEN ALL WORDS ARE USED	HASH	21
	SA1	A1+B6	X1=NEXT G WORD	HASH	22
	bX3	X1-X3	EXCLUSIVE OF	HASH	23
	JP	NEXTW		HASH	24
COSY	MX0	30	X0=MASK	HASH	25
	BX4	X3		HASH	26
	LX4	30	X4=LEFT HALF OF X3	HASH	27
	BX4	X3-X4	X4=EXCL OR OF EACH HALF OF X3	HASH	28
	bK4	-X0*X4	X4=LOW ORDER 33 BITS OF EXCL OR	HASH	29
	SA2	B3	X2=NTAB	HASH	30
	PX0	X4	X0=DIVIDEND PACKED	HASH	31
	PX2	X2	PACK X2	HASH	32
	NX3	X2	NORMALIZE X2 INTO X3	HASH	33
	FX6	X0/X3	DIVIDE	HASH	34
	UX6	B7,X6	UNPACK	HASH	35
	LX6	B7,X6	SHIFT	HASH	36
	PX6	K6	PACK X6	HASH	37
	DX6	X6*X2	REMULTIPLY	HASH	38
	UX6	X6	UNPACK X6	HASH	39
	IX6	X4-X6	X6=REMAINDER=HASH CODE	HASH	40
	SX6	X6+B6	SUBSCRIPT FORM OF HASH CODE,	HASH	41
	SA6	B4	STORE RESULT IN KODE,	HASH	42
	JP	HASH		HASH	43
	END			HASH	44

```

C      INTEGER FUNCTION IDEAL (KH)
C      THIS FUNCTION IS USED TO CLASSIFY A GIVEN CHARACTER AS TO WHETHER IT
C      IS ALPHABETIC, NUMERIC, ONE OF THE SPECIAL OR PUNCTUATION CHARACTERS
C      RECOGNIZED BY FAMULUS, OR ALL OTHER CHARACTERS (PERMITTED IN FAMULUS
C      TEXT BUT NOT LISTED IN CONTROLS),
C      THE CHARACTER IS KH, AN INTEGER BETWEEN 3 AND 63. IT WILL NORMALLY BE
C      OBTAINED FROM THE KHAR FUNCTION.
C      THE RESULT IS AN INTEGER GIVING THE CLASS INTO WHICH KH FALLS, AS
C      FOLLOWS.
C
C      IDEAL(KH)  KH
C      1         BLANK
C      2         ALPHABETIC (A-Z)
C      3         NUMERIC (0-9)
C      4         /
C      5         (
C      6         )
C      7         ,
C      8         -
C      9         *
C     10         : (OR)
C     11         & (AND)
C     12         % (NOT)
C     13         ALL OTHERS
C
C      a PURPOSE OF THIS FUNCTION IS TO CONVERT TO ONE PLACE MOST OF THE
C      MACHINE TYPE VARIATIONS ARISING FROM THE VARIETY OF CHARACTER CODES THAT
C      EXISTS AMONG MACHINES.
C      WHEN CONVERTING TO A OTHER MACHINE, THE ONLY CHANGE NECESSARY IS IN
C      THE DATA STATEMENT, ASSUMING A RANGE OF 0-63 IN KH. IF KH HAS A
C      DIFFERENT RANGE, THE INTEGER STATEMENT MUST ALSO BE CHANGED.
C      THIS DATA STATEMENT IS FOR THE CONTROL DATA 6400/660 ,
C      INTEGER CODE(64)
C      DATA (CODE(J),J=1,64)/13,26*2,10*3,13,8,9,4,5,6,13,13,1,7,7*13,10,
C      11,6*13,12,13/
C      -----
C      CODE ( 4 ) = CODE ( 63 )
C      64 IS UNUSABLE CHARACTER. IT IS USED AS A NOT CHARACTER
C      THIS CHANGE WAS NECESSARY BECAUSE UT-200 TERMINAL CANNOT SEND
C      A NOT CHARACTER WHICH IS DISPLAY CODE 768
C      -----
C      K=KH+1
C      IDEAL=CODE(K)
C      RETURN
C      END

```

```

IDEAL 2
IDEAL 3
IDEAL 3
IDEAL 4
IDEAL 6
IDEAL 7
IDEAL 8
IDEAL 9
IDEAL 10
IDEAL 11
IDEAL 12
IDEAL 13
IDEAL 14
IDEAL 15
IDEAL 16
IDEAL 17
IDEAL 18
IDEAL 19
IDEAL 20
IDEAL 21
IDEAL 22
IDEAL 23
IDEAL 24
IDEAL 25
IDEAL 26
IDEAL 27
IDEAL 28
IDEAL 29
IDEAL 30
IDEAL 31
IDEAL 32
IDEAL 33
IDEAL 34
IDEAL 35
IDEAL 36
IDEAL 37
IDEAL 38
IDEAL 39
IDEAL 40
IDEAL 41
IDEAL 42

```


C PRINTS THE ITEMS THAT ARE IN A PREFACE RECORD.	IDPRIN 3
C T IS A LITERAL OF NT CHARACTERS TO PRINT AS A LABEL.	IDPRIN 4
COMMON /FAM/SLASH,STAR,COMMA,ZERO,MXCHAR,MXDES,MXCHW,	IDPRIN 5
MXFRE,MXWFRE	IDPRIN 6
INTEGER SLASH,STAR,COMMA,ZERO	IDPRIN 7
INTEGER ID(2),FLD(2),IFLD(2),FL(10),T(2)	IDPRIN 8
CALL PAGER(-1)	IDPRIN 9
MIXED=0	IDPRIN10
DO 10 L=1,NFLD	IDPRIN11
K=IFLD(L)	IDPRIN12
IF(K.NE.L) MIXED=1	IDPRIN13
10 FL(L)=FLD(K)	IDPRIN14
NNT=(NT+MXCHW-1)/MXCHW	IDPRIN15
CALL PAGER(5)	IDPRIN16
PRINT 1000, (T(I),I=1,NNT)	IDPRIN17
1000 FORMAT(19H IDENTIFICATION OF , 10A10)	IDPRIN18
PRINT 2000, (ID(J),J=1,MXFRE)	IDPRIN19
2000 FORMAT(1X,12A8,A4)	IDPRIN20
PRINT 3000, (FLD(K),K=1,NFLD)	IDPRIN21
3000 FORMAT(11H FIELDS ARE,2X,10(A4,2X))	IDPRIN22
IF(MIXED.NE.0) PRINT 3001, (FL(K),K=1,NFLD)	IDPRIN23
3001 FORMAT(8H ORDERED,5X,10(A4,2X))	IDPRIN24
IF(KDES.EQ.0) GO TO 20	IDPRIN25
PRINT 4000,FLD(KDES)	IDPRIN26
4000 FORMAT(26H THE DESCRIPTOR FIELD IS ,A4)	IDPRIN27
20 RETURN	IDPRIN28
END	IDPRIN29

SUBROUTINE INCARD	INCARD 2
C READS ALL INPUT CARDS,	INCARD 3
C A CALL TO INCARD READS THE NEXT CARD INTO INPT COMMON AND DOES THE	INCARD 4
C NECESSARY BOOKKEEPING.	INCARD 5
C THE CARD IMAGE IS STORED IN PACKED FORM, NCHW CHARACTERS PEY CELL,	INCARD 6
C TESTS FOR END OF FILE IN CARD INPUT,	INCAHC 7
COMMON /CORE/NCHW,BLANKS,BLANK	INCARD 8
INTEGER BLANKS,BLANK	INCARD 9
COMMON /EDIN/INTAPE	INCARD10
COMMON /FAM/SLASH,STAR,COMMA,ZERO,MXCHAR,MXCODES,MXWDES,MXCHW,	INCARD11
MXFREE,MXWFRE	INCARD12
INTEGER SLASH,STAR,COMMA,ZERO	INCARD13
COMMON /INPT/CARD(10),KOLUM,HOLD,ENDCAR,BLACAR,CONCAR,ERRIN,NWC,	INCARD14
MXFL,MXCAR,NCARIN,KHR,KIN,LIT,LITX,SYM(5),NUHS,NCSYM,CATLOG,	INCARD15
KARA,KARB	INCARD16
INTEGER CARD,SYM	INCARD17
LOGICAL HOLD,ENDCAR,BLACAR,CONCAR,ERRIN,CATLOG	INCARD18
LOGICAL MATCH	INCARD19
CALL BUTLER	INCARD20
IF(CATLOG) GO TO 100	INCARD21
CALL DIVEST	INCARD22
IF(ENDCAR) GO TO 12	INCARD23
READ(INTAPE,1000) (CARD(J),J=1,NWC)	INCARD24
1000 FORMAT(10A8)	INCARD25
IF (EOF (INTAPE)) 10,20	JRT 56
C END OF FILE,	INCARD27
10 ENDCAR=.TRUE.	INCARD28
HOLD=.FALSE.	INCARD29
KOLUM=0	INCARD30
BLACAR=.FALSE.	INCARD31
CONCAR=.TRUE.	INCARD32
GO TO 500	INCARD33
C ROT END OF FILE,	INCARD34
20 ENDCAR=.FALSE.	INCARD35
HOLD=.TRUE.	INCARD36
KOLUM=0	INCARD37
NCARIN=NCARIN+1	INCARD38
BLACAR=.FALSE.	INCARD39
DO 30 J=1,NWC	INCARD40
IF(.NOT.MATCH(CARD(J),BLANKS)) GO TO 40	INCARD41
30 CONTINUE	INCARD42
BLACAR=.TRUE.	INCARD43
40 CONCAR=KHAR(CARD,1).EQ.SLASH	INCARD44
GO TO 500	INCARD45
C THE REMAINDER OF INCARD IS SOLELY FOR COMPATIBILITY WITH CATALOG, AN	INCARD46
C OBSOLETE PREDECESSOR SF FAMULUS. IT IS ENTERED ONLY WHEN CITATION	INCARD47
C CARDS ARE READ.	INCARD48
100 KARA=7	INCARD49
KARB=72	INCARD50
IF(ENDCAR) GO TO 110	INCARD51
IF(BLACAR) GO TO 120	INCARD52
READ(INTAPE,1000) (CARD(J),J=1,NWC)	INCARD53
IF (EOF (INTAPE)) 110,130	JRT 57
C END OF FILE.	INCARD55
113 ENDCAR=.TRUE.	INCARD56
HOLD=.FALSE.	INCARD57
KOLUM=0	INCARD58

BLACAR=.FALSE.	INCARD59
CONCAR=.TRUE.	INCARD60
GO TO 500	INCARD61
C BLANK CARD	INCARD62
120 BLACAR=.FALSE.	INCARD63
GO TS 141	INCARD64
C NOT END OF FILE.	INCARD65
130 KR=KHAR(CARD,79)	INCARD66
BLACAR=(KR.EQ.BLANK.OR.KR.EQ.ZERO).AND.KHAR(CARD,80).EQ.ZERO+1	INCARD67
NCARIN=NCARIN+1	INCARD68
140 END CAR=.FALSE.	INCARD69
HOLD=.TRUE.	INCARD70
KOLUM=0	INCARD71
CONCAR=.FALSE.	INCARD72
500 CONTINUE	INCARD73
CALL TALMED(4HCARD,CARD,0,80)	INCARD74
CALL SPOT(NCARIN,BLACAR,CONCAR,ENDCAR,4HILL,	INCARD75
, 24HNCARINBLACARCONCARENDCAR)	INCARD76
RETURN	INCARD77
END	INCARD78

SUBROUTINE INGEST(CIT, LAST)	INGEST 2
C READS A CITATION FROM CARDS, IN SPECIAL FORMAT, FOR THE PURPOSE OF	INGEST 3
C USING IT IN LOGGING.	INGEST 4
C IF THE CITATION CONSISTS ONLY OF A CARD WITH * IN COLUMN 1, INGEST	INGEST 5
C SETS LAST=1 AND RETURNS.	INGEST 6
COMMON /CORE/NCHW, BLANKS, BLANK	INGEST 7
INTEGER BLANKS, BLANK	INGEST 8
COMMON /FIELD/FIELD(10), IFIELD(10), NFIELD, KDESCR	INGEST 9
INTEGER FIELD	INGEST 10
COMMON /IPT/CARD(10), KOLIM, HOLD, ENDCAR, BLACAR, CONCAR, ERRIN, NWC,	INGEST 11
, MXFL, MAXCAR, NCARIN, KHR, KIN, LIT, LITX, SYM(5), NUNG, NCSYM, CATLOG,	INGEST 12
, KARA, KARB	INGEST 13
INTEGER IARR, SYM	INGEST 14
LOGICAL FLD, ENDCAR, BLACAR, CONCAR, ERRIN, CATLOG	INGEST 15
INTEGER IIT(3,2), TEXT(500), FLD, DOLL	INGEST 16
LIMIT(N) = ((1+NCHW-1)/NCHW)*NCHW	INGEST 17
CALL BUTLE,	INGEST 18
CALL PAGE(-2)	INGEST 19
NCHAR=0	INGEST 20
DOLL=KHA(1H*,1)	INGEST 21
LC=3*NFIELD*NCHA	INGEST 22
CALL INCARD	INGEST 23
DO 50 K=1, NFIELD	INGEST 24
NT=0	INGEST 25
NART=0	INGEST 26
FLD=BLANKS	INGEST 27
CALL CHOP(CARD, 1, FLD, J, 4)	INGEST 28
KF=LOOK(FIELD, IFIELD, FLD)	INGEST 29
IF (KF.GT.8) GO TO 40	INGEST 30
IF (KF.LT.8) CALL HALT	INGEST 31
IF (KHA(CARD, 1), #, DOLL) CALL HALT	INGEST 32
LT=0	INGEST 33
C READ FIELD TEXT.	INGEST 34
20 CALL INCARD	INGEST 35
IF (KHA(CARD, 1), #, DOLL) GO TO 30	INGEST 36
CALL CHOP(CARD, 0, TEXT, LT, 80)	INGEST 37
LT=LT+80	INGEST 38
GO TO 20	INGEST 39
C FINISHED READING TEXT FOR CURRENT FIELD.	INGEST 40
30 NT=LT	INGEST 41
CALL SCOUT(1, 0, BLANK, TEXT, 0, NT, K)	INGEST 42
NT=NT-K	INGEST 43
CALL CHOP(TEXT, 0, CIT, LC, LIMIT(NT))	INGEST 44
NCHAR=NCHAR+NT	INGEST 45
CALL ARTY(TEXT, 0, NT, NART)	INGEST 46
NCHAR=NCHAR+NT	INGEST 47
CALL ARTY(TEXT, 0, NT, NART)	INGEST 48
40 KF=K	INGEST 49
CIT(1, KF)=NT	INGEST 50
CIT(2, KF)=LC	INGEST 51
CIT(3, KF)=NART	INGEST 52
LC=LC+LIMIT(NT)	INGEST 53
50 CONTINUE	INGEST 54
C DONE.	INGEST 55
LAST=1	INGEST 56
IF (NCHAR, LE, 0) RETURN	INGEST 57
DO 60 K=1, NFIELD	INGEST 58
60 CALL TALMOD(FIELD(KF), CIT, CIT(2, KF), CIT(1, KF))	INGEST 59
LAST=0	INGEST 60
RETURN	INGEST 61
END	INGEST 62

SUBROUTINE IOID(IO,MT,IDD,FLD,IFLD,KDES)	IOID	2
C READS AND WRITES THE PREFACE RECORD OF TAPE,	IOID	3
C IO=0 FOR READING, IO=1 FOR WRITING.	IOID	4
C MT IS THE TAPE UNIT 0.	IOID	5
COMMON /FAM/SLASH,STAR,COMMA,ZERO,MXCFAR,MXCODES, XWDES,MXCHW,	IOID	6
MXFREE,MXWFERE	IOID	7
INTEGER SLASH,STAR,COMMA,ZERO	IOID	8
INTEGER IDD(2),FLD(2),IFLD(2)	IOID	9
CALL BUTLER	IOID	10
CALL SPOT(IO,MT,NFLD,KDES,4HIIII,24HIC MT NFLD KDES)	IOID	11
IX=MXWFERE	IOID	12
CALL REWIND(MT)	IOID	13
IF(IO.EQ.1) GO TO 50	IOID	14
C READING...	IOID	15
READ(MT)	IOID	16
IF (EOF (MT)) 20,30	JRT	58
C END OF FILE.	IOID	18
20 CALL PAGER(-2)	IOID	19
CALL PAGER(1)	IOID	20
PRINT 1000, MT	IOID	21
1000 FORMAT(5H TAPE,12,17H WAS NOT MOUNTED - 104(1H*))	IOID	22
STOP	IOID	23
C NOT END OF FILE.	IOID	24
30 CALL REWIND(MT)	IOID	25
READ(MT)(IDD(J),J=1,NW),NFLD,(FLD(J),J=1,NFLD),(IFLD(J),J=1,NFLD),	IOID	26
.,KDES	IOID	27
GO TO 100	IOID	28
C WRITING...	IOID	29
50 WRITE(MT)(IDD(J),J=1,NW),NFLD,(FLD(J),J=1,NFLD),(IFLD(J),J=1,NFLD),	IOID	30
.,KDES	IOID	31
100 CONTINUE	IOID	32
CALL SPOT(IO,MT,NFLD,KDES,4HIIII,24HIC MT NFLD KDES)	IOID	33
CALL TALMOD(3HIDD,IDD,0,MXFREE)	IOID	34
RETURN	IOID	35
END	IOID	36

SUBROUTINE IOTAPE(IO,MT,CIT,LAST)	IOTAPE 2
C THIS SUBROUTINE PERFORMS ALL INPUT AND OUTPUT OF CITATIONS TO AND FROM	IOTAPE 3
C TAPE. IT DOES NOT DO THE PREFACE RECORD, NOR DOES IT HANDLE DISK	IOTAPE 4
C OPERATIONS.	IOTAPE 5
C MT IS THE FORTRAN TAPE UNIT NUMBER.	IOTAPE 6
C	IOTAPE 7
C IF IO=0, WE ARE READING...	IOTAPE 8
C A CITATION IS READ FROM TAPE MT INTO THE ARRAY CIT.	IOTAPE 9
C SETS LAST=1 IF THE CITATION IS A DUMMY CITATION INDICATING END OF	IOTAPE10
C FILE. OTHERWISE, SETS LAST=0.	IOTAPE11
C	IOTAPE12
C IF IO=1, WE ARE WRITING...	IOTAPE13
C A CITATION LOCATED AT CIT IN CORE IS WRITTEN ONTO TAPE MT AS ONE	IOTAPE14
C LOGICAL RECORD, IF LAST=0.	IOTAPE15
C IF LAST=1, HOWEVER, WE DO NOT WRITE INFORMATION THAT IS IN CIT BUT	IOTAPE16
C INSTEAD WE WRITE A DUMMY RECORD INDICATING END OF FILE.	IOTAPE17
COMMON /CORE/NCHW,BLANKS,BLANK	IOTAPE18
INTEGER BLANKS,BLANK	IOTAPE19
COMMON/FAM/SLASH,STAR,COMMA,ZERO,MXCHAR,MXCDIS,MXWDES,MXCHW,	IOTAPE20
MXFREE,MXWFRE	IOTAPE21
INTEGER SLASH,STAR,COMMA,ZERO	IOTAPE22
COMMON /FIEL/FIELD(10),IFIELD(10),NFIELD,KDESCR	IOTAPE23
INTEGER FIELD	IOTAPE24
COMMON /TEST/ITEST	IOTAPE25
INTEGER CIT(2)	IOTAPE26
DATA MXWCIT,NIN,NOUT/3*0/	IOTAPE27
C NIN=THE NUMBER OF TIMES A CITATION HAS BEEN READ	IOTAPE28
C NOUT=THE NUMBER OF TIMES A CITATION HAS BEEN WRITTEN.	IOTAPE29
CALL BUTLER	IOTAPE30
IF (MXWCIT,EG.0)MXWCIT=3*NFIELD+(MXCHAR+NCHW-1)/NCHW+NFIELD	IOTAPE31
MTX=MT	IOTAPE32
IF(IO.EQ.1) GO TO 20	IOTAPE33
C READING...	IOTAPE34
NIN=NIN+1	IOTAPE35
IF (IO.EQ.-1) GO TO 50	IOTAPE36
READ(MT)N,(CIT(J),J=1,N)	IOTAPE37
IF(N.EQ.1) GO TO 10	IOTAPE38
LAST=0	IOTAPE39
CALL SPOT(IO,MT,LAST,N,4HIIII,24HI0 MT LAST N)	IOTAPE40
CALL KORAN(6HCIT***,CIT,3*NFIELD)	IOTAPE41
LC=3*NFIELD*NCHW	IOTAPE42
NC=100	IOTAPE43
CALL TALMOD(3HCIT,CIT,LC,NC)	IOTAPE44
IF (N.LE.0.OR.N.GT.MXWCIT.OR.MT.NE.MTX) CALL HALT	IOTAPE45
IF(ITEST.NE.0) CALL PRNCIT(CIT,-0,0,128)	IOTAPE46
GO TO 100	IOTAPE47
C END OF FILE READING.	IOTAPE48
10 LAST=1	IOTAPE49
CALL SPOT(IO,MT,LAST,N,4HIIII,24HI0 MT LAST N)	IOTAPE50
GO TO 100	IOTAPE51
C WRITING...	IOTAPE52
20 IF(LAST.EQ.1) GO TO 30	IOTAPE53
NOUT=NOUT+1	IOTAPE54
K=3*NFIELD-2	IOTAPE55
N=(CIT(K)+CIT(K+1)+NCHW-1)/NCHW	IOTAPE56
CALL SPOT(IO,MT,LAST,N,4HIIII,24HI0 MT LAST N)	IOTAPE57
WRITE(MT)N,(CIT(J),J=1,N)	IOTAPE58

IF (ITEST.NE.0) CALL PRNCIT(CIT,-1,0,128)	IOTAPE59
GO TO 100	IOTAPE60
C END OF FILE WRITING.	IOTAPE61
30 N=1	IOTAPE62
N=-1	IOTAPE63
CALL SPOT(10,MT,LAST,0,4HIIII,24H10 MT LAST N)	IOTAPE64
WRITE(MT),N	IOTAPE65
GO TO 100	IOTAPE66
C SKIPPING	IOTAPE67
50 READ(MT) N	IOTAPE68
NIN=NIN+1	IOTAPE69
IF (N.EQ.1) GO TO 10	IOTAPE70
LAST=0	IOTAPE71
100 CONTINUE	IOTAPE72
CALL SPOT(NIN,NOUT,0,0,4HIIII,24HNIN NOUT)	IOTAPE73
RETURN	IOTAPE74
END	IOTAPE75

SUBROUTINE JUST(X,LX,NX,NSUB,ADD)	JUST	2
C LEFT-JUSTIFY A STRING OF CHARACTERS AND FILLS OUT THE LAST CELL WITH	JUST	3
C BLANKS.	JUST	4
X IS A CHARACTER STRING, THE SUBSTRING OF X A IS JUSTIFIED: RUNS	JUST	5
C FROM POSITION LX+1 TO LX+NX INCLUSIVE.	JUST	6
C SETS NSUB AS THE NUMBER OF LEADING AND TRAILING BLANK CHARACTERS	JUST	7
C REMOVED FROM THE SUBSTRING.	JUST	8
C AFTER THIS IS DONE, THE LAST CELL IS FILLED OUT WITH BLANKS IF NOT	JUST	9
C ALREADY FULL. NADD IS SET AS THE NUMBER OF BLANKS SO ADDED.	JUST	10
C IF X IS LESS THAN 1, SETS NSUB=NADD=0 AND RETURNS.	JUST	11
COMMON / CREZCHX, BLANKS, BLANK	JUST	12
INTEGER BLANKS, BLANK	JUST	13
INTEGER X(2)	JUST	14
NADD=0	JUST	15
NSUB=0	JUST	16
IF(NX.LT.1) RETURN	JUST	17
NSUB=NX	JUST	18
CALL SCOUT(0,0,BLANK,X,LX,NX,N)	JUST	19
N=N-N	JUST	20
IF(N.GT.0) CALL CHOP(X,LX+N,X,LX,N)	JUST	21
CALL SCOUT(1,0,BLANK,X,LX,N,N)	JUST	22
N=N-N	JUST	23
NSUB=NX-	JUST	24
JB=LX+N	JUST	25
NADD=((JB+NCHW-1)/NCHW)*NCHW-JB	JUST	26
CALL CHOP(BLANKS,0,X,JB,NADD)	JUST	27
RETURN	JUST	28
END	JUST	29

SUBROUTINE KORAN(TAG,JUG,NJUG)	KORAN 2
C THIS IS A DE BUGGING SUBROUTINE, USED TO PRINT THE CONTENTS OF A ONE-	KORAN 3
C DIMENSIONAL INTEGER ARRAY.	KORAN 4
C THE ARRAY IS JUG, ITS LENGTH IS NJUG.	KORAN 5
C TAG IS AN 8-CHARACTER LABEL TO PRINT WITH IT.	KORAN 6
C IN CASE NJUG IS 0, ONLY TAG IS PRINTED.	KORAN 7
C PRINTS IN 7-CHARACTER FIELDS.	KORAN 8
COMMON /TEST/ITEST	KORAN 9
INTEGER TAG(1),JUG(2)	KORAN 10
,ATA MAX/15/	KORAN 11
IF(ITEST,EQ,0) RETURN	KORAN 12
IMINO(1,00,NJUG)	KORAN 13
IF(N,GE,1) GO TO 10	KORAN 14
CALL PAGER(1)	KORAN 15
PRINT 1000, TAG	KORAN 16
1000 FORMAT(1X,A8)	KORAN 17
RETURN	KORAN 18
11 CALL PAGER((IN+MAX-1)/MAX)	KORAN 19
IF(J,NE,MAX) PRINT 2000, TAG,(JUG(J),J=1,N)	KORAN 20
IF(N,EQ,MAX) PRINT 3000, TAG,(JUG(J),J=1,N)	KORAN 21
2000 FORMAT(1X,A8,2X,5I7,3X,5I7,3X,5I7/(11X,5I7,3X,5I7,3X,5I7))	KORAN 22
3000 FORMAT(1X,A8,2X,5I7,3X,5I7,3X,5I7)	KORAN 23
IF(N,GE,NJUG) GO TO 100	KORAN 24
PRINT 400,	KORAN 25
4000 FORMAT(17X,11H,...,ETC,...) -	KORAN 26
CALL PAGER(0)	KORAN 27
100 RETURN	KORAN 28
END	KORAN 29

```

      INTEGER FUNCTION LEADER(CITA,CITB)
C COMPARES TWO CITATIONS, CITA AND CITB, FOR ALPHABETICAL ORDER,
C THE RESULTS ARE INDICATED AS FOLLOWS,...
C   LEADER=-1    CITA PRECEDES CITB
C   LEADER=0     CITA IS TIED WITH CITB
C   LEADER=1     CITB PRECEDES CITA
C THE NUMBER AND ORDER OF THE FIELDS ARE OBTAINED FROM COMMON,
C LEADING ARTICLES (A, AN, THE) IN FIELDS ARE IGNORED.
      COMMON /FIEL/FIELD(10),IFIELD(10),NFIELD,KDESCR
      INTEGER FIELD
      INTEGER CITA(3,2),CITB(3,2)
      DO 10 KF=1,NFIELD
      K=IFIELD(KF)
      LA=CITA(2,K)+CITA(3,K)
      NA=CITA(1,K)-CITA(3,K)
      LB=CITB(2,K)+CITB(3,K)
      NB=CITB(1,K)-CITB(3,K)
      LEAD=LEXICO(CITA,LA,NA,CITB,LB,NB)
      IF(LEAD.NE.0) GO TO 20
10  CONTINUE
20  LEADER=LEAD
      RETURN
      END

```

```

LEAUEH 2
LEADER 3
LEADER 4
LEADER 5
LEADER 6
LEADER 7
LEADER 8
LEADER 9
LEADER10
LEADER11
LEADER12
LEADER13
LEADER14
LEADER15
LEADER16
LEADER17
LEADER18
LEADER19
LEADER20
LEADER21
LEADER22
LEADER23
LEADER24

```

INTEGER FUNCTION LEXICO(X,LX,IX,Y,LY,RY)	LEXICO 2
C COMPARES TWO SUBSTRINGS FOR ALPHABETICAL ORDER.	LEXICO 3
C THE SUBSTRINGS ARE X, HAVING NX CHARACTERS WITH LX POINTING TO THE	LEXICO 4
C FIRST SUBSTRING CHARACTER, AND Y, HAVING NY CHARACTERS WITH LY	LEXICO 5
C POINTING TO THE FIRST ONE.	LEXICO 6
C THE VALUE OF LEXICO INDICATES THE RESULT AS FOLLOWS...	LEXICO 7
C LEXICO=-1, X PRECEDES Y.	LEXICO 8
C LEXICO=0, X AND Y ARE IDENTICAL.	LEXICO 9
C LEXICO=1, Y PRECEDES X.	LEXICO10
C ASSUMES BOTH X AND Y ARE GREATER THAN 0.	LEXICO11
C IN ORDER TO ALPHABETIZE WE ESTABLISH THE FOLLOWING (ASCENDING) ORDER	LEXICO12
C AMONG THE 63 POSSIBLE CHARACTERS...	LEXICO13
C BLANK	LEXICO14
C , , ( ) + - * / \$ = # [ ] " _ ! & ' ? < > @ ! % ;	LEXICO15
C A THROUGH Z	LEXICO16
C 0 THROUGH 9	LEXICO17
C THE ORDERING WITHIN THE PUNCTUATION AND SPECIAL CHARACTERS IS SOMEWHAT	LEXICO18
C ARBITRARY.	LEXICO19
C FOR SPEED, IT IS ADVISABLE TO HAND CODE THIS SUBROUTINE.	LEXICO20
INTEGER X(2),Y(2),LEX(64)	LEXICO21
C THE FOLLOWING STATEMENT DEPENDS ON THE MACHINE CHARACTER CODE.	LEXICO22
DATA (LEX(J),J=1,64)/63,27,28,29,30,31,32,33,34,35,36,37,38,39,40,	LEXICO23
41,42,43,44,45,46,47,48,49,50,51,52,53,54,55,56,57,58,59,60,61,62,	LEXICO24
5,6,7,8,3,4,9,10,0,2,1,11,12,13,14,15,16,17,18,19,20,21,22,23,24,	LEXICO25
52,26/	LEXICO26
IF(NX+NY.EQ.0) GO TO 15	LEXICO27
IF(NX.EQ.0) 20 TO 40	LEXICO28
IF(NY.EQ.0) GO TO 30	LEXICO29
N=MIN0(NX,NY)	LEXICO30
DO 10 J=1,N	LEXICO31
KX=KHAR(X,LX+J)	LEXICO32
KY=KHAR(Y,LY+J)	LEXICO33
IF(KX.NE.KY) GO TO 20	LEXICO34
10 CONTINUE	LEXICO35
IF(NX.LT.1Y) GO TO 30	LEXICO36
IF(NY.LT.1X) GO TO 40	LEXICO37
15 LEXICO=0	LEXICO38
RETURN	LEXICO39
20 IF(LEX(KX+1).GT.LEX(KY+1)) GO TO 40	LEXICO40
30 LEXICO=-1	LEXICO41
RETURN	LEXICO42
40 LEXICO=1	LEXICO43
RETURN	LEXICO44
END	LEXICO45

IDENT	LOOK.	LOOK(U,N,V)	LOOK	2
LIST	G,H,M		LOOK	3
* LOOK(U,N,V)	INTEGER FUNCTION		LOOK	5
* THE GIVEN N-WORD ARRAY U IS SEARCHED FOR AN			LOOK	6
* EXACT COPY OF V (ALL BITS THE SAME).			LOOK	7
* IF ONE IS FOUND, LOOK RETURNS THE CORRESPONDING SUBSCRIPT			LOOK	8
* OF U (OR THE FIRST ONE, IF MORE THAN ONE IS FOUND).			LOOK	9
* IF NOT FOUND, LOOK RETURNS N+1.			LOOK	10
* N MAY BE 0, IN WHICH CASE LOOK ALWAYS			LOOK	11
* RETURNS 1.			LOOK	12
* 6400 TIME = 4*L MICROSECONDS, WHERE L IS RESULT (APPROX.)			LOOK	13
* THE FOLLOWING IS AN EQUIVALENT FORTRAN FUNCTION (WHERE MATCH			LOOK	14
* IS A LOGICAL FUNCTION HAVING THE VALUE TRUE IF			LOOK	15
* ITS TWO ARGUMENTS AGREE IN ALL BITS, FALSE			LOOK	16
* OTHERWISE).			LOOK	17
* INTEGER FUNCTION LOOK(U,N,V)			LOOK	18
* INTEGER I(N)			LOOK	19
* L=1			LOOK	20
* IF(N.EQ.0) GO TO 30			LOOK	21
* DO 10 J=1,N			LOOK	22
* 10 IF(MATCH(U(J),V)) GO TO 20			LOOK	23
* J=N+1			LOOK	24
* 20 L=J			LOOK	25
* LOOK=L			LOOK	26
* RETURN			LOOK	27
* END			LOOK	28
4			LOOK	29
* PROGRAMMED BY ROBERT M. RUSSELL, DECEMBER 1967.			LOOK	30
*			LOOK	31
ENTRY LOOK			LOOK	32
VFD 24/4HLOOK,36/3			LOOK	33
LOOK BSS 1			LOOK	34
SB1 X1	MODIFICATION TO ALLOW THIS ROUTINE	JRT	59	
SA2 A1+1	TO WORK ON FTN COMPILER	JRT	60	
SB2 X2	INSTEAD OF RUN COMPILER	JRT	61	
SA2 A1+2		JRT	62	
SB3 X2		JRT	63	
SA1 B2		LOOK	35	
SB2 X1		LOOK	36	
NZ B2,GO		LOOK	37	
SX6 1		LOOK	38	
JP LOOK		LOOK	39	
GO SA1 B3		LOOK	40	
SB4 a0		LOOK	41	
SB5 1		LOOK	42	
NEXT SA2 B1+B4		LOOK	43	
BX3 X1-X2		LOOK	44	
SB4 B4+B5		LOOK	45	
ZR X3,TEST		LOOK	46	
NE 64,B2,NEXT		LOOK	47	
FLAT SX6 B2+1		LOOK	48	
JP LOOK		LOOK	49	
TEST PL X3,FOUND		LOOK	50	
NE B4,B2,NEXT		LOOK	51	
JP FLAT		LOOK	52	
FOUND SX6 B4		LOOK	53	
JP LOOK		LOOK	54	
		LOOK	55	
END				

	IDENT MATCH.	MATCH(G,H) LOGICAL FUNCTION		MATCH	
* MATCH(G,H)	LOGICAL FUNCTION			MATCH	4
* RETURNS THE VALUE ,TRUE, IF G AND H ARE IDENTICAL WORDS, HIT FOR BIT,				MATCH	5
* RETURNS THE VALUE ,FALSE, IF G AND H DIFFER IN ONE OR MORE BITS,				MATCH	6
* ,FALSE, IS REPRESENTED BY 60 ZERO BITS AND ,TRUE, BY 60 ONE BITS,				MATCH	7
* 6400 TIME = 8 MICROSECONDS (APPROX,,),				MATCH	8
* PROGRAMMED BY ROBERT M. RUSSELL, DECEMBER 1967.				MATCH	9
	ENTRY WATCH			MATCH	10
	VFD 30/5HMATCH,30/2			MATCH	11
MATCH	BSS 1			MATCH	12
	SB1 X1	MODIFICATION TO ALLOW THIS ROUTINE	JRT	64	
	SA2 A1+1	TO WORK ON FTN COMPILER	JRT	65	
	SB2 X2	INSTEAD OF RUN COMPILER	JRT	66	
	SA2 A1+2		JRT	67	
	SB3 X2		JRT	68	
	SA1 B1		MATCH	13	
	SA2 B2		MATCH	14	
	BX1 X1-X2		MATCH	15	
	SX6 B0		MATCH	16	
	NG X1,MATCH		MATCH	17	
	NZ X1,MATCH		MATCH	18	
	BX6 -Xb		MATCH	19	
	JP MATCH		MATCH	20	
	END		MATCH	21	

LOGICAL FUNCTION MEMBER(LST,MAX,ITM)	MEMBER 2
C SEARCHES A LIST OF CITATION NUMBERS IN STANDARD FAMULUS FORM, TO	MEMBER 3
C DETERMINE IF A GIVEN CITATION NUMBER IS IN THE LIST.	MEMBER 4
C AN EXAMPLE OF SUCH A LIST IS ONE PUNCHED AS FOLLOWS, (3,5-7,10,12-15)	MEMBER 5
C THE LIST IS IN THE ARRAY LST. THE EXAMPLE WOULD BE STORED THUS, 3,5,-7	MEMBER 6
C ,10,12,-15,0.	MEMBER 7
C THE NUMBER TO BE LOOKED UP IS ITM.	MEMBER 8
C SETS MEMBER=TRUE IF ITM IS IN THE LIST.	MEMBER 9
C SETS MEMBER=FALSE IF IT IS NOT.	MEMBER 10
C IT IS PERMISSABLE THAT THE LIST BE EMPTY, I.E., LST(1)=0.	MEMBER 11
C MAX IS THE MAXIMUM NUMBER OF ENTRIES IN LST.	MEMBER 12
INTEGER LST(2)	MEMBER 13
IF(MAX.LE.0) GO TO 20	MEMBER 14
MEMBER=.TRUE.	MEMBER 15
DO 10 J=1,MAX	MEMBER 16
NOW=ABS(LST(J))	MEMBER 17
IF(NOW.EQ.0) GO TO 20	MEMBER 18
IF(LST(J).GT.0) LAST=NOW	MEMBER 19
IF(ITM.GE.LAST.AND.ITM.LE.NOW) GO TO 30	MEMBER 20
10 LAST=NOW	MEMBER 21
20 MEMBER=.FALSE.	MEMBER 22
30 RETURN	MEMBER 23
END	MEMBER 24

	IDENT	MOVE,	MOVE(U,V,N)	MOVE	2
	LIST	G,E,M		MOVE	3
*	MOVE(U,V,N)			MOVE	5
*	IF N IS POSITIVE, SETS V(J)=U(J) FOR J=1,N.			MOVE	6
*	IF N IS NEGATIVE, SETS V(J)=U FOR J=1,-N.			MOVE	7
a	IF N=0, NO ACTION			MOVE	8
*	THE EQUIVALENT FORTRAN PROGRAM IS			MOVE	9
*	SUBROUTINE MOVE(U,V,N)			MOVE	10
*	INTEGER U(N),V(N)			MOVE	11
*	IF(N.EQ.0) RETURN			MOVE	12
*	IF(N.LT.0) GO TO 20			MOVE	13
*	DO 10 J=1,N			MOVE	14
*	10 V(J)=U(J)			MOVE	15
*	RETURN			MOVE	16
a	20 N=-N			MOVE	17
*	DO 30 J=1,M			MOVE	18
*	30 V(J)=U(1)			MOVE	19
a	RETURN			MOVE	20
a	END			MOVE	21
*	6400 TIME FOR N POSITIVE = 4.6 *N MICROSECONDS (APPROX)			MOVE	22
*	6400 TIME FOR N NEGATIVE = 2.8 *N MICROSECONDS (APPROX)			MOVE	23
*				MOVE	24
*	PROGRAMMED BY ROBERT M. RUSSELL, DECEMBER 1967.			MOVE	25
*				MOVE	26
	ENTRY	MOVE		MOVE	27
	VFL	24/4HMOVE,36/3		MOVE	28
MOVE	BSS	1		MOVE	29
	SB1	X1	MODIFICATION TO ALLOW THIS ROUTINE	JRT	69
	SA2	A1+1	TO WORK ON FTR COMPILER	JRT	70
	SB2	X2	INSTEAD OF RUN COMPILER	JRT	71
	SA2	A1+2		JRT	72
	SB3	X2		JRT	73
	SA1	B3	X1=N	MOVE	30
	SB3	X1	B3=N	MOVE	31
	ZR	B3,MOVE		MOVE	32
	SB4	a0	B4=0=INCREMENT=J-1	MOVE	33
	SB5	1	B5=1=CONSTANT	MOVE	34
	NG	B3,ONE	TEST FOR NEGATIVE N	MOVE	35
ALL	SA1	B1+B4	X1=U(J)	MOVE	36
	BX6	X1	X6=U(J)	MOVE	37
	SA6	B2+B4	STORE U(J) IN V(J)	MOVE	38
	SB4	B4+B5	J=J+1	MOVE	39
	NE	B4,B3,ALL	TEST J=N	MOVE	40
	JP	MOVE		MOVE	41
ONE	SA1	B1	X1=U	MOVE	42
	BX6	X1	X6=U	MOVE	43
	SH3	B0-B3		MOVE	44
NEXT	SA6	B2+B4	STORE U IN V(J)	MOVE	45
	SB4	B4+B5	J=J+1	MOVE	46
	NE	B4,B3,NEXT	TEST J=N	MOVE	47
	JP	MOVE		MOVE	48
	END			MOVE	49

SUBROUTINE NIMROD(KF,CIT)	NIMROD 2
C THIS SUBROUTINE SEARCHES A FIELD FOR RETRIEVAL TERMS.	NIMROD 3
COMMON /FAM/SLASH,STAR,COMMA,ZERO,MXCHAR,MXDESC,MXWLES,MXCHW,	NIMROD 4
MXFRL,MXWFR	NIMROD 5
INTEGER SLASH,STAR,COMMA,ZERO	NIMROD 6
COMMON /FIEL/FIELD(10),IFIELD(10),NFIELD,KDESCR	NIMROD 7
INTEGER FIELD	NIMROD 8
COMMON TEXT(250),LTEXT(50),NCTEXT(50),NTEXT,	NIMROD 9
JTERM(60),MODFLD(60),LOCATE(60),NTERM,MAYBE(64),KODET(64),	NIMROD10
KUKE(150),KKUKE,LUKE(150),MLUKE,LIST(150),MLIST,	NIMROD11
KSERCH(10),NSERCH,KFLD(10),MXTEXT,MXCTXT,MXTERM,MXLUKE	NIMROD12
INTEGER CIT(3,2)	NIMROD13
CALL BUTLER	NIMROD14
L=CIT(2,KF)	NIMROD15
N=CIT(1,KF)	NIMROD16
10 CALL SCAN(1,KODET,CIT,L,N,NSKIP,KH)	NIMROD17
L=L+NSKIP	NIMROD18
N=N-NSKIP	NIMROD19
IF(MAYBE(KH+1),EQ,0) 20 TO 30	NIMROD20
DO 20 J=1,NTERM	NIMROD21
JX=JTERM(J)	NIMROD22
CALL SIMILE(0,TEXT,LTEXT(JX),NCTEXT(JX),CIT,L,N,LIKE)	NIMROD23
IF(LIKE,EQ,0) GO TO 20	NIMROD24
LOCATE(J)=1	NIMROD25
CALL TALMOD(6HINIMROD,TEXT,LTEXT(JX),NCTEXT(JX))	NIMROD26
20 CONTINUE	NIMROD27
30 IF(KF,EQ,KDESCR) CALL SCOUT(0,1,COMMA,CIT,L,N,NSKIP)	NIMROD28
IF(KF,NE,KDESCR) CALL SCAN(0,KODET,CIT,L,N,NSKIP,KH)	NIMROD29
L=L+NSKIP	NIMROD30
N=N-NSKIP	NIMROD31
IF(N,GT,0) GO TO 10	NIMROD32
RETURN	NIMROD33
C METHOD.	NIMROD34
C DEFINE THE SET OF 36 CHARACTERS T=(A-Z AND 0-9).	NIMROD35
C DEFINE THE SET OF CHARACTERS X=(ALL FIRST-CHARACTERS OF TERMS).	NIMROD36
C THE SEARCH ALGORITHM IS AS FOLLOWS.	NIMROD37
C 10 SKIP CIT CHARACTERS UNTIL ONE IS FOUND THAT IS IN T.	NIMROD38
C IF IT IS NOT IN X, GO TO 30.	NIMROD39
C COMPARE EACH TERM WITH CIT AT THAT POSITION.	NIMROD40
C SET LOCAT(J)=1 IF J-TH TERM MATCHES CIT.	NIMROD41
C 30 IF KF=KDESCR, SKIP CIT CHARACTERS UNTIL A COMMA IS	NIMROD42
C FOUND.	NIMROD43
C IF KF=KDESCR, SKIP CIT CHARACTERS UNTIL ONE IS	NIMROD44
C FOUND THAT IS NOT IN T.	NIMROD45
C GO TO 10.	NIMROD46
C	NIMROD47
END	

SUBROUTINE NOAH(KERR)	NOAH	2
C READS VOCABULARY FROM CARDS AND STORES IT IN USABLE FORM FOR	NOAH	3
C PROCESSING.	NOAH	4
C THESE VARIABLES IN BLANK COMMON ARE SET...	NOAH	5
C DICT,LDICT,NDICT,NVOC,NTRIV,HASH.	NOAH	6
C THESE VARIABLES IN BLANK COMMON ARE USED...	NOAH	7
C MXHASH,MXENT,MXDICT,WORD.	NOAH	8
C KERR INDICATES ERROR CONDITIONS. KERR=1, TOO MANY ENTRIES, KERR=2,	NOAH	9
C TOO MANY CHARACTERS IN DICTIONARY. KERR=0, NO ERROR.	NOAH	10
COMMON /CORE/NCHW,BLANKS,BLANK	NOAH	11
INTEGER BLANKS,BLANK	NOAH	12
COMMON /FAM/SLASH,STAR,COMMA,ZERO,MXCHAR,MXCDL,MXWDL,MXCHW,	NOAH	13
MXFREE,MXWFE	NOAH	14
INTEGER SLASH,STAR,COMMA,ZERO	NOAH	15
COMMON /INPT/CARD(10),KOLUM,HOLD,ENDCAR,BLACAR,CONCAR,ERRIN,NWC,	NOAH	16
MXFL,MXCAR,NCARIN,KHR,KIN,LIT,LITX,SYM(5),NUMB,NCSYM,CATLOG,	NOAH	17
KARA,KARB	NOAH	18
INTEGER CARD,SYM	NOAH	19
LOGICAL HOLD,ENDCAR,BLACAR,CONCAR,ERRIN,CATLOG	NOAH	20
COMMON /TEST/ITEST	NOAH	21
COMMON NDICT,NVOC,NTRIV,MXHASH,MXENT,MXDICT,WORD(5),	NOAH	22
DICT(6750),LDICT(6501),HASH(9511)	NOAH	23
INTEGER DICT,HASH,WORD,KRANK(6500)	NOAH	24
EQUIVALENCE(KRANK,HASH)	NOAH	25
LOGICAL GOING	NOAH	26
KERR=0	NOAH	27
GOING=.FALSE.	NOAH	28
IF(KHR(CARD,1).EQ.SLASH) CALL DIVEST	NOAH	29
CALL MOVE(0,HASH,-MXHASH)	NOAH	30
C COLLECT NEXT ENTRY.	NOAH	31
10 CALL MOVE(BLANKS,WORD,-MXWDL)	NOAH	32
KTRIV=1	NOAH	33
LC=0	NOAH	34
NC=0	NOAH	35
C PROCESS NEXT CHARACTER.	NOAH	36
20 CONTINUE	NOAH	37
IF(GOING) CALL GETCH	NOAH	38
GOING=.TRUE.	NOAH	39
IF(ENDCAR.OR.KHR.EQ.SLASH.AND.KOLUM.EQ.1) GO TO 50	NOAH	40
IF(KHR.EQ.COMMA) GO TO 40	NOAH	41
IF(KHR.EQ.STAR) GO TO 35	NOAH	42
IF(KIN.GT.3.AND.KIN.LE.8) GO TO FG4	NOAH	43
30 IF(LC.GE.MXCDL.OR.LC.EQ.0.AND.KHR.EQ.BLANK) GO TO 20	NOAH	44
LC=LC+1	NOAH	45
CALL CHAR(KHR,WORD,LC)	NOAH	46
IF(KHR.NE.BLANK) NC=LC	NOAH	47
GO TO 20	NOAH	48
35 KTRIV=-1	NOAH	49
40 IF(NC.EQ.0) GO TO 10	NOAH	50
50 IF(NC.EQ.0) GO TO 70	NOAH	51
C INSERT ENTRY INTO DICTIONARY. (IF ALREADY THERE, OVERRIDE IT.)	NOAH	52
CALL SCAT(WORD,0,NC,DICT,LDICT,NDICT,HASH,MXHASH,JDICT,JHASH)	NOAH	53
IF(JDICT.LE.NDICT) GO TO 60	NOAH	54
HASH(JHASH)=JDICT	NOAH	55
IF(JDICT.GT.MXENT) GO TO 901	NOAH	56
LDT=IABS(LDICT(JDICT))	NOAH	57
LDICT(JDICT+1)=LDT+NC	NOAH	58

IF(LDICT(JDICT+1).GT.MXDICT) GO TO 902	NOAH	59
CALL TALMOD(4HWORD,WORD,0,MXCLES)	NOAH	60
CALL CHOP(WORD,0,DICT,LDI,NC)	NOAH	61
NDICT=NDICT+1	NOAH	62
60 LDICT(JDICT)=ISIGN(LDICT(JDICT),KTRIV)	NOAH	63
IF(.NOT.(ENDCAR.OR.KHR.EQ.SLASH)) GO TO 10	NOAH	64
C DONE.	NOAH	65
70 NTRIV=0	NOAH	66
IF(NDICT.EQ.0) GO TO 903	NOAH	67
DO 80 K=1,NDICT	NOAH	68
IF(ISIGN(1,LDICT(K)).LT.0) NTRIV=NTRIV+1	NOAH	69
80 CONTINUE	NOAH	70
NVOC=NDICT-NTRIV	NOAH	71
HOLD=,TRUE.	NOAH	72
CALL SPOT(NDICT,NVOC,NTRIV,0,4HIIII,24HNDICT NVOC NTRIV)	NOAH	73
CALL KORAN(5HLDICT,LDICT,NDICT+1)	NOAH	74
DO 90 J=1,NDICT	NOAH	75
LD=IABS(LDICT(J))	NOAH	76
ND=IABS(LDICT(J+1))-LD	NOAH	77
90 CALL TALMOD(4HDICT,DICT,LD,ND)	NOAH	78
100 RETURN	NOAH	79
C ERRORS.	NOAH	80
901 KERR=1	NOAH	81
CALL PAGER(2)	NOAH	82
PRINT 9001	NOAH	83
9001 FORMAT(/37H TOO MANY ENTRIES IN INPUT DICTIONARY,91(1H*))	NOAH	84
GO TO 100	NOAH	85
902 KERR=2	NOAH	86
CALL PAGER(2)	NOAH	87
PRINT 9002	NOAH	88
9002 FORMAT(/40H TOO MANY CHARACTERS IN INPUT DICTIONARY,88(1H*))	NOAH	89
GO TO 100	NOAH	90
903 KERR=3	NOAH	91
CALL DIVEST	NOAH	92
CALL PAGER(2)	NOAH	93
PRINT 9003	NOAH	94
9003 FORMAT(/31H WHERE IS YOUR INPUT DICTIONARY,97(1H*))	NOAH	95
STOP	NOAH	96
904 KERR=4	NOAH	97
CALL PAGER(3)	NOAH	98
PRINT 9004, KOLUN,CARD	NOAH	99
9004 FORMAT(/28H ILLEGAL CHARACTER IN COLUMN,13,16H OF THIS CARD...	NOAH	100
, 10A8/1X,127(1H*))	NOAH	101
STOP	NOAH	102
END	NOAH	103

```

SUBROUTINE OFFPAGE ( N )
  IF ( N .EQ. 0 )      PRINT 300 , "1"
  IF ( N .LE. 1 )      PRINT 300 , " "
300 FORMAT ( A8 )
C.....LDSET(SURST=PPGER-OFFPAGE).....
C... THE LDSET CARD IS USED TO TURN OFF PAGE FORMATTING ON PRINTOUT
C... THIS ROUTINE TAKES THE PLACE OF *PAGER* WHICH NORMALLY PAGE FORMATS
      RETURN
END

```

SUBROUTINE PAGER( )	06/12/71
C (FAMULUS VERSION)	PAGER 3
C PAGER COMPLETELY HANDLES PAGE AND LINE BOOKKEEPING.	A R 4
C IF N IS GREATER THAN J, PAGER ASSUMES N LINES ARE ABOUT TO BE	PAGER 5
C PRINTED AND CALCULATES WHETHER THERE ARE N LINES LEFT ON CURRENT	PAGER 6
C PAGE. IF THERE ARE NOT, PAGER SPACES PRINTER TO THE TOP OF THE NEXT	PAGER 7
C PAGE AND PRINTS THE STANDARD PAGE HEADING. IF THERE ARE, PAGER SIMPLY	PAGER 8
C UPDATES THE LINE COUNT BY N.	PAGER 9
C IF N IS 0, PAGER OPERATES AS IF N WERE A LARGE NEGATIVE NUMBER.	PAGER 10
C IF N IS LESS THAN 0, THE PRINTER WILL SKIP -N LINES (BUT NOT PASS THE	PAGER 11
C TOP OF THE FOLLOWING PAGE).	PAGER 12
C THE STANDARD PAGE HEADING CONSISTS OF 100 CHARACTERS IN PAGHD, THE	PAGER 13
C DATE, AND THE PAGE NUMBER.	PAGER 14
C THE ITEMS IN PACING COMMON ARE AUTOMATICALLY INITIALIZED AT THE FIRST	PAGER 15
C CALL, INCLUDING THE SETTING OF PAGHD TO BLANKS. IF OTHER MATERIAL	PAGER 16
C IS DESIRED IN PAGHD, PUT IT THERE AFTER THE FIRST CALL.	PAGER 17
C PAGER CALLS A SYSTFY SUBROUTINE, DATE, TO SUPPLY THE DATE FOR THE	PAGER 18
C HEADING LINE.	PAGER 19
C	PAGER 20
C COMMON /PAGING/ CONTAINS THE FOLLOWING ITEMS.,	PAGER 21
C THE CURRENT LINE NUMBER IS IPAGE.	PAGER 22
C THE NUMBER OF LINES LEFT ON THE CURRENT PAGE IS LINES,	PAGER 23
C THE MAXIMUM NUMBER OF LINES ON A PAGE IS MAXLIN.	PAGER 24
C THE 100 CHARACTERS TO BE PRINTED IN THE HEADING LINE ARE IN PAGHD.	PAGER 25
C	PAGER 26
C PROGRAMMED BY ROBERT M. RUSSELL, FEBRUARY, 1968.	PAGER 27
C	PAGER 28
COMMON /PAGING/IPAGE,LINES,MAXLIN,PAGHD(13)	PAGER 29
DATA IPAGE,LINES,MAXLIN,(PAGHD(J),J=1,13)/0,0,53,13*1H /	PAGER 30
DATA ITCH,DAT/0,1H /	PAGER 31
IF(ITCH.NE.0) GO TO 5	PAGER 32
ITCH=1	PAGER 33
CALL DATE(DAT)	PAGER 34
C NEXT STATEMENT IS FOR LC PERKFLEY COMPUTER CENTER ONLY. IT MOVES	PAGER 35
C THE PRINT PAGE TO FIRST LINE BEYOND NEXT CONCAVE PAPER FOLD,	PAGER 36
PRINT 4000	PAGER 37
4000 FORMAT(1H6)	PAGER 38
5 CONTINUE	PAGER 39
IF(N.EQ.0) GO TO 50	PAGER 40
IF(N.GT.0) GO TO 30	PAGER 41
NN=-N	PAGER 42
LINES=LINES-NN	PAGER 43
IF(LINES.LE.0) GO TO 50	PAGER 44
10 IF(NN.LE.1) GO TO 20	PAGER 45
PRINT 2000	PAGER 46
2000 FORMAT(1H0)	PAGER 47
,N=NN-2	PAGER 48
GO TO 10	PAGER 49
20 IF(NN.EQ.1) PRINT 1000	PAGER 50
1000 tORMAT(1x)	PAGER 51
GO TO 100	PAGER 52
30 LINES=LINES-N	PAGER 53
IF(LINES.GE.0) GO TO 100	PAGER 54
IPAGE=IPAGE+1	PAGER 55
PRINT 3000, (PAGHD(J),J=1,13),DAT,IPAGE	PAGER 56
3000 FORMAT(1H1,12A8,A4,9X,A10,18/1H0)	PAGER 57
LINES=MAX(0,MAXLIN-N)	PAGER 58
GO TO 100	PAGER 59
50 LINES=0	PAGEH 60
100 RETURN	PAGER 61
END	PAGER 62

SUBROUTINE POLISH	POLISH 2
C READS A BOOLEAN SEARCH REQUEST FORMULA FROM INPUT CARDS, CONVERTS ITS	POLISH 3
C SYMBOLS INTO USEFUL FORM, AND PRODUCES A REVERSE POLISH STRING	POLISH 4
C EQUIVALENT TO THE FORMULA,	POLISH 5
COMMON /CORE/NCHW,BLANKS,PLANK	POLISH 6
INTEGER BLANKS,BLANK	POLISH 7
COMMON /DSCRPT/IDSCRPT	POLISH 8
COMMON /FIEL/FIELD(10),IFIELD(10),NFIELD,KDESCR	POLISH 9
INTEGER FIELD	POLISH10
COMMON /INPT/CARD(10),KOLUM,HOLD,ENDCAR,BLACAR,CONCAR,ERRIN,NWC,	POLISH11
MXFL,MAXCAR,NCARIN,KHR,KIN,LIT,LITX,SYM(5),NUMB,NCSYM,CATLOG,	POLISH12
KARA,KARB	POLISH13
INTEGER CARD,SYM	POLISH14
LOGICAL HOLD,ENDCAR,BLACAR,CONCAR,ERRIN,CATLOG	POLISH15
COMMON TEXT(250),LTEXT(50),NCTEXT(50),NTEXT,	POLISH16
JTERM(60),MODFLD(60),LOCATE(60),NTERM,MAYBE(64),KODET(64),	POLISH17
KUKE(150),NKUKE,LUKE(150),NLUKE,LIST(150),NLIST,	POLISH18
KSERCH(10),NSERCH,KFLD(10),MXTEXT,MXCTXT,MXTERM,MXLUKE	POLISH19
INTEGER KEY(12)	POLISH20
DATA (KEY(J),J=1,12)/3*0,-4,-5,3*0,-1,-2,-3,0/	POLISH21
CALL BUTLER	POLISH22
JT=0	POLISH23
KU=0	POLISH24
LT=0	POLISH25
CALL MOVE(0,KFLD,-NFIELD)	POLISH26
CALL MOVE(0,MAYBE,-64)	POLISH27
IDSCRPT=1	POLISH28
10 LITOLD=LITX	POLISH29
CALL SYMBOL	POLISH30
LIT=LIT	POLISH31
LITX=LITX	POLISH32
C TEST FOR SYMBOL PRECEDENCE ERROR,	POLISH33
KE=1	POLISH34
GO TO (12,900,900, 14, 16,900,900,900, 161 161 12, 16),LIT	POLISH35
12 GO TO (900,900, 20, 20,900,900,900,900, 20, 20, 20,900),LITX	POLISH36
14 GO TO (20,900, 20, 20,900,900,900,900, 20, 20, 20,900),LITX	POLISH37
16 GO TO (20,900,900,900,20,900,900,900,900,900,900,900),LITX	POLISH38
20 IF(LIT.NE.1) GO TO 40	POLISH39
C ALPHABETIC-TYPE SYMBOL,	POLISH40
NCS=NCSYM	POLISH41
IF(LITOLD.EQ.1.AND.LITX.EQ.4) GO TO 30	POLISH42
C DESCRIPTOR,	POLISH43
IF(JT.LE.0) GO TO 24	POLISH44
DO 22 J=1,JT	POLISH45
N=NCTEXT(J)	POLISH46
IF(N.NE.NCS) GO TO 22	POLISH47
CALL SIMILE(1,TEXT,LTEXT(J),N,SYM,0,NCS,LIKE)	POLISH48
IF(LIKE.EQ.1) GO TO 26	POLISH49
22 CONTINUE	POLISH50
24 KE=2	POLISH51
IF(JT.GE.MXTEXT) GO TO 900	POLISH52
KE=3	POLISH53
IF(LT+NCS.GT.MXCTXT) GO TO 900	POLISH54
JT=JT+1	POLISH55
CALL CHOP(SYM,0,TEXT,LT,NCS)	POLISH56
MAY=1+KHAF(SYM,1)	POLISH57
MAYBE(MAY)=1	POLISH58

LTEXT(JT)=LT	POLISH59
NCTEXT(JT)=NCS	POLISH60
LT=LT+NCS	POLISH61
J=JT	POLISH62
26 KUK=J	POLISH63
GO TO 50	POLISH64
C FIELD LABEL.	POLISH65
30 KE=4	POLISH66
IF(NCS.GT.MXFL) GO TO 900	POLISH67
K=LOOK(FIELD,NFIELD,SYM)	POLISH68
KE=5	POLISH69
IF(K.GT.NFIELD) GO TO 900	POLISH70
KUK=K	POLISH71
KFLD(K)=1	POLISH72
GO TO 50	POLISH73
C OPERATOR, PARENTHESES, OR END-OF-STATEMENT.	POLISH74
40 KUK=KEY(LIT)	POLISH75
C STASH OPERATOR OR OPERAND IN KUK.	POLISH76
50 KE=6	POLISH77
IF(KU.GE.MXLUKE) GO TO 900	POLISH78
KU=KU+1	POLISH79
KUKE(KU)=KUK	POLISH80
NKUK=KU	POLISH81
60 IF(KUK.NE.0) GO TO 10	POLISH82
C FINALIZE SYMBOL COLLECTION.	POLISH83
NKUK=KU	POLISH84
NTEXT=JT	POLISH85
CALL KORAN(5HMAYBE,MAYBE,64)	POLISH86
CALL KORAN(5HLTEXT,LTEXT,NTEXT)	POLISH87
CALL KORAN(6HNCTEXT,NCTEXT,NTEXT)	POLISH88
CALL KORAN(4HKFLD,KFLD,NFIELD)	POLISH89
CALL KORAN(4HKUKE,KUKE,NKUK)	POLISH90
DO 65 JT=1,NTEXT	POLISH91
65 CALL TALMUD(4HTEXT,TEXT,LTEXT(JT),NCTEXT(JT))	POLISH92
C CONSTRUCT JTERM AND MODFLD ARRAYS, AND REVISE KUK IN ORDER TO COMBINE	POLISH93
C FIELD MODIFIERS WITH DESCRIPTORS, AND ALSO TO CHANGE DESCRIPTOR	POLISH94
C INLIGIES IN KUK SO THEY POINT TO JTERM AND MODFLD INSTEAD OF TO TEXT	POLISH95
C ARRAY.	POLISH96
JTR=0	POLISH97
KK=0	POLISH98
KU=0	POLISH99
70 KU=KU+1	POLIS100
KUK=KUKE(KU)	POLIS101
IF(KUK.LE.0) GO TO 95	POLIS102
MF=0	POLIS103
IF(KUKE(KU+1).NE.-4) GO TO 75	POLIS104
MF=KUKE(KU+2)	POLIS105
KU=KU+3	POLIS106
75 JTXT=KUK	POLIS107
IF(JTR.EQ.0) GO TO 85	POLIS108
DO 80 J=1,JTR	POLIS109
IF(JTERM(J).EQ.JTXT.AND.MODFLD(J).EQ.MF) GO TO 90	POLIS110
80 CONTINUE	POLIS111
85 KE=7	POLIS112
IF(JTR.GE.MXTERM) GO TO 900	POLIS113
JTR=JTR+1	POLIS114
JTERM(JTR)=JTXT	POLIS115

MODFLD(JTR)=MF	POLIS116
J=JTR	POLIS117
90 KUK=J	POLIS118
95 KK=KK+1	POLIS119
KUKE(KK)=KUK	POLIS120
IF(KU,LT,NKUKE) GO TO 70	POLIS121
NTERM=JTR	POLIS122
NKUKE=KK	POLIS123
CALL KORAN(4HKUKE,KUKE,NKUKE)	POLIS124
CALL KORAN(5HJTERM,JTERM,NTERM)	POLIS125
CALL KORAN(6HMODFLD,MODFLD,NTERM)	POLIS126
KE=12	POLIS127
IF(NTERM,LT,1) GO TO 900	POLIS128
C CONVERT THE FORMULA IN KUKE TO A REVERSE POLISH STRING IN LUKE.	POLIS129
110 NLIST=0	POLIS130
NLUKE=0	POLIS131
LOP=0	POLIS132
DO 175 KU=1,NKUKE	POLIS133
K=KUKE(KU)	POLIS134
IF(K,EQ,0) GO TO 180	POLIS135
IF(K,LT,0) GO TO 120	POLIS136
C OPERAND,	POLIS137
NLUKE=NLUKE+1	POLIS138
LUKE(NLUKE)=K	POLIS139
LOP=LOP+1	POLIS140
GO TO 170	POLIS141
C NON-OPERAND,	POLIS142
120 KK=IABS(K)	POLIS143
GO TO (140,140,140,130,160),KK	POLIS144
C LEFT PARENTHESIS	POLIS145
130 NLIST=NLIST+1	POLIS146
LIST(NLIST)=KK	POLIS147
GO TO 170	POLIS148
C OPERATOR,	POLIS149
145 IF(NLIST,LE,0) GO TO 150	POLIS150
LST=LIST(NLIST)	POLIS151
IF(LST,EQ,4,OR,LST,LT,KK) GO TO 150	POLIS152
NLUKE=NLUKE+1	POLIS153
LUKE(NLUKE)=-LST	POLIS154
IF(LST,LE,2) LOP=LOP-1	POLIS155
KE=13	POLIS156
IF(LOP,LE,0) GO TO 900	POLIS157
CALL KORAN(5HLIST1,LIST,NLIST)	POLIS158
NLIST=NLIST-1	POLIS159
GO TO 140	POLIS160
150 NLIST=NLIST+1	POLIS161
LIST(NLIST)=KK	POLIS162
GO TO 170	POLIS163
C RIGHT PARENTHESIS,	POLIS164
160 KE=8	POLIS165
IF(NLIST,LE,0) GO TO 900	POLIS166
LST=LIST(NLIST)	POLIS167
CALL KORAN(5HLIST2,LIST,NLIST)	POLIS168
NLIST=NLIST-1	POLIS169
IF(LST,EQ,4) GO TO 170	POLIS170
NLUKE=NLUKE+1	POLIS171
LUKE(NLUKE)=-LST	POLIS172

IF(LST.LE.2) LOP=LOP-1	POLIS173
KE=9	POLIS174
IF(LOP.LE.0) GO TO 900	POLIS175
GO TO 160	POLIS176
170 CONTINUE	POLIS177
175 CONTINUE	POLIS178
C KUKE IS DEPLETED, MUST NOW EMPTY LIST INTO LUKE.	POLIS179
180 IF(NLIST.LE.0) GO TO 190	POLIS180
LST=LIST(NLIST)	POLIS181
CALL KORAN(5HLIST3,LIST,NLIST)	POLIS182
NLIST=NLIST-1	POLIS183
KE=10	POLIS184
IF(LST.EQ.4) GO TO 900	POLIS185
NLUKE=NLUKE+1	POLIS186
LUKE(NLUKE)=-LST	POLIS187
IF(LST.LE.2) LOP=LOP-1	POLIS188
GO TO 180	POLIS189
C LUKE IS FULL,	POLIS190
190 KE=11	POLIS191
IF(LOP.NE.1) GO TO 900	POLIS192
NLUKE=NLUKE+1	POLIS193
LUKE(NLUKE)=0	POLIS194
CALL KORAN(4HLUKE,LUKE,NLUKE)	POLIS195
C NORMAL RETURN,	POLIS196
IDSCRIP=0	POLIS197
RETURN	POLIS198
C ERROR ANALYSIS,	POLIS199
900 IF(KOLUM.NE.1, AND, KHR.NE,SLASH) CALL DIVEST	POLIS200
CALL PAGER(2)	POLIS201
GO TO (908,902,903,904,904,	POLIS202
906,906,907,908,907,	POLIS203
908,908,908),KE	POLIS204
902 PRINT 9002	POLIS205
9002 FORMAT(15H TOO MANY TERMS,5X,108(1H*))	POLIS206
GO TO 930	POLIS207
903 PRINT YOU3	POLIS208
9003 FORMAT(31H TOO MANY CHARACTERS I?FORMULA,5X,92(1H*))	POLIS209
GO TO 935	POLIS210
904 PRINT 9004, SYM	POLIS211
9004 FORMAT(29H THIS IS NOT A FIELD LABEL,,,5A8,5X,54(1H*))	POLIS212
GO TO 930	POLIS213
906 PRINT 9006	POLIS214
9006 FORMAT(28H TOO MANY SYMBOLS IN FORMULA,5X,95(1H*))	POLIS215
GO TO 930	POLIS216
907 PRINT 9007	POLIS217
9007 FORMAT(25H PARENTHESES OC NOT MATCH,5X,98(1H*))	POLIS218
GO TO 930	POLIS219
908 PRINT 9008	POLIS220
9008 FORMAT(18H INCORRECT FORMULA,5X,105(1H*))	POLIS221
C FINALIZE ERROR ANALYSIS.	POLIS222
930 PRINT 9050,KE	POLIS223
9050 FORMAT(118X,6HPOLISH I4)	POLIS224
CALL PAGER(-1)	POLIS225
CALL KORAN(4HKUKE,KUKE,NKUKE)	POLIS226
CALL KORAN(4HLUKE,LUKE,NLUKE)	POLIS227
CALL KORAN(4HLIST,LIST,NLIST)	POLIS228
ERRIN=.TRUE.,	POLIS229
IDSCRIP=0	POLIS230
RETURN	POLIS231
END	POLIS232

SUBROUTINE PRETTY(CIT,LCIT,INDENT,LEN,FCHOI,JLAB,NCHOI)	07/18/75
C PRINTS A CITATION IN BIBLIOGRAPHIC FORMAT,	Pretty 3
C THE CITATION IS AT CIT,	Pretty 4
C THE CITATION NUMBER IS LCIT, WHICH IS PRINTED ON THE LEFT UNLESS IT IS	Pretty 5
C 0 IN WHICH CASE IT IS NOT PRINTED,	Pretty 6
C INDENT=1 SAYS TO INDENT ALL LINES BUT THE FIRST,	Pretty 7
C INDENT=0 SAYS TO INDENT NONE OF THEM,	Pretty 8
C THE FIELDS ARE PRINTED IN THE ORDER DICTATED BY THE IFIELD ARRAY,	Pretty 9
C LEN IS THE PAGE WIDTH, IN CHARACTERS, IT IS THE MAX. NUMBER OF	Pretty 10
C CITATION TEXT CHARACTERS ON A LINE PLUS 4 CHARACTERS FOR LCIT(IF LCIT	Pretty 11
C IS NOT 0),	Pretty 12
COMMON /CORE/NCHW,BLANKS,BLANK	Pretty 13
INTEGER BLANKS,BLANK	Pretty 14
COMMON /FIEL/FIELD(10),IFIELD(10),NFIELD,KDESCR	Pretty 15
INTEGER FFIELD	Pretty 16
INTEGER CIT(3,2),LINE(16)	Pretty 17
C THE DIMENSION OF LINE() IS LENLIN+1.	Pretty 18
LOGICAL GOLINE,GOCHAR,ENTROP,BLOCK,DONE	Pretty 19
LOGICAL FCHOI	07/16/76
INTEGER JLAB(10)	07/18/77
DATA LENLIN/15/	Pretty 20
C INITIALIZE	Pretty 21
GOLINE=,FALSE,	Pretty 22
KF=0	Pretty 23
NC=0	Pretty 24
LL=0	Pretty 25
ENTROP=LCIT,EQ,0	Pretty 26
BLOCK=INDENT,LE,J	Pretty 27
MOST=MIN0(LEN-7,LENLIN*NCHW)	Pretty 28
IF(ENTROP) MOST=LEN	Pretty 29
DONE=,FALSE,	Pretty 30
CALL PAGER(-1)	Pretty 31
C PREPARE TO START A NEW LINE,	Pretty 32
10 CALL MOVE(BLANKS,LINE,-LENLIN)	Pretty 33
GOCHAR=,FALSE,	Pretty 34
NL=MOST-LL	Pretty 35
C TEST IF CURRENT FIELD DEPLETED AND IF SO, GET NEXT FIELD,	Pretty 36
20 IF(NC,GT,0) GO TO 30	Pretty 37
IF(FCHOI) GO TO 25	07/16/78
IF(KF,GE,NFIELD) GO TO 80	Pretty 38
KF=KF+1	Pretty 39
KFF=IFIELD(KF)	Pretty 40
GO TO 27	07/18/79
25 IF(KF,GE,NCHOI) GO TO 80	07/18/10
YF=KF+1	07/18/11
KFF=JLAB(KF)	07/18/12
27 NC=CIT(1,KFF)	07/18/13
LC=CIT(2,KFF)	Pretty 42
IF(,NOT,(GOCHAR,AND,NC,GT,0)) GO TO 40	Pretty 43
N=MIN0(NL,3)	Pretty 44
CALL CHOP(BLANKS,0,LINE,LL,-N)	Pretty 45
LL=LL+N	Pretty 46
NL=NL-N	Pretty 47
GO TO 20	Pretty 48
C SELECT GROUP GF CHARS TO MOVE INTO LINE WAITING TO PRINT,	Pretty 49
30 CALL SCOUT(0,0,BLANK,CIT,LC,NC,M)	Pretty 50
LC=LC+M	Pretty 51

NC=NC-M	PRETTY52
N=MIN0(NC,NL+1)	PRETTY53
IF(NC,LE,IL) GO TO 35	PRETTY54
CALL SCOUT(1,1,BLANK,CIT,LC,N,M)	PRETTY55
IF(M,EQ,N,AND, .NOT,GOCHAR) M=1	PRETTY56
N=N-M	PRETTY57
35 IF(N,LE,0) GO TO 40	PRETTY58
CALL CHOP(CIT,LC,LINE,LL,N)	PRETTY59
LC=LC+N	PRETTY60
NC=NC-N	PRETTY61
LL=LL+N	PRETTY62
NL=NL-N	PRETTY63
GOCHAR=,TRUE.	PRETTY64
GO TO LO	PRETTY65
C PRINT THE LINE WAITING.	PRETTY66
40 CALL PAGER(1)	PRETTY67
IF(ENTROP) GO TO 60	PRETTY68
IF(GOLINE) 60 TO 50	PRETTY69
C---PRINT FIRST LINE WITH CITATION NUMBER	07/18/14
GOLINE=,TRUE.	PRETTY70
PRINT 1000, LCIT,(LINE(J),J=1,LENLIN)	PRETTY71
C---CHANGE SPACING TO ALLOW MORE MARGIN ON LEFT (MAX WIDTH = 132)	57/18/15
1000 FORMAT (5X,14,1H,,2X,15A8)	07/18/16
GO TO 70	PRETTY73
C---PRINT REMAINING LINES UNDER COMMON FORMAT	07/18/17
50 PRINT 2000, (LINE(J),J=1,LENLIN)	PRETTY74
2000 FORMAT (12X,15A8)	07/18/18
GO TO 70	PRETTY76
C---PRINT LINE WITHOUT CITATION NUMBER WITH FIRST LINE OFFSET = RJ OLSON	07/18/19
60 PRINT 3000, (LINE(J),J=1,LENLIN)	PRETTY77
ENTROP = ,FALSE.	07/18/20
GOLINE = ,TRUE.	07/18/21
3000 FORMAT(8X,15A8)	67/16/22
C RESET FOR NEXT LINE, IF ANY.	PRETTY79
70 IF(DONE) GO TO 103	PRETTY80
LL=5	PRETTY81
IF(BLOCK) LL=0	PRETTY82
GO TO 10	PRETTY83
C FINISHED WITH CITATION, BUT MUST PRINT PARTIAL LAST LINE.	PRETTY84
60 DONE=,TRUE.	PRETTY85
GO TO 40	PRETTY86
C ALL DONE, RETURN.	PRETTY87
100 RETURN	PRETTY88
END	PRETTY89

<pre> SUBROUTINE PRNCIT(CIT,LCIT,JCIT, LEN) C PRINTS A CITATION IN A NICE FORMAT. C THE CITATION IS LOCATED AT CIT. C LCIT IS THE CITATION NUMBER, PRINTED ON THE LEFT MARGIN. C LEN IS THE PAGE WIDTH, IN CHARACTERS. C JCIT IS ANOTHER CITATION NUMBER (FOR INSTANCE, ITS NUMBER IN A C PREVIOUS RUN) WHICH PRNCIT PRINTS ON THE RIGHT, PROVIDING IT IS C GREATER THAN 0. IF PRINTED, IT FALLS OUTSIDE THE LEN LIMIT (UNLIKE C LCIT). C INTERNAL FORTRAN VARIABLES ARE... C LINE(14) ONE LINE OF TEXT, NCHW CHARACTERS PER CELL. C ALLOWANCE IS MADE FOR A 128-CHARACTER PRINTER. C FTAG LABEL OF CURRENT FIELD. C TOP INDICATES WHETHER FIRST LINE IS THE CURRENT LINE. C MOST MAX. NO. OF TEXT CHARS PER LINE. C KF CURRENT FIELD. C LEFT NO. OF CHARS REMAINING TO PRINT, CURRENT FIELD. C JC CHAY POSITION, REL TO CIT(1), 3F LAST C CHAR PRINTED SO FAR IN CURRENT FIELD. C NC NO. OF TEXT CHARS BEING PRINTED IN CURRENT LINE. C NWL NO. OF CELLS REQUIRED FOR ONE LINE SF TEXT. C COMMON /CORE/NCHW,BLANKS,BLANK INTEGER BLANKS,BLANK COMMON /FAM/SLASH,STAR,COMMA,ZERO,MXCHAR,MXCODES,MXWDES,MXCHW, MXFREE,MXWFRE INTEGER SLASH,STAR,COMMA,ZERO COMMON /FIEL/FIELD(10),IFIELD(10),NFIELD,KUESCR INTEGER FIELD INTEGER CIT(3,2),LINE(14),FTAG LOGICAL TOP CALL PAGER(-1) TOP=.TRUE. MOST=MAX0(10,MIN0(Len-12,109)) NWL=14 PO 30 K=1,NFIELD KF=IFIELD(K) LEFT = CIT(1,KF) IF(LEFT.LT. 1) GO TO 30 JC=CIT(2,KF) FTAG=FIELD(KF) IF(LEFT.GT.MXCHAR) CALL HALT IF(JC.LE.0) CALL HALT 10 CALL SCOUT(0,0,BLANK,CIT,JC,LEFT,M) JC=JC+M LEFT=LEFT-M IF(LEFT.LT.1) GO TO 30 NC=LEFT IF(NC .LE. MOST) GO TO 20 NC=MOST+1 CALL SCOUT(1,1,BLANK,CIT,JC,NC,M) NC=NC-M CALL SCOUT(1,0,BLANK,CIT,JC,NC,M) NC=NC-M IF(NC.LE.0) NC=MOST 20 CALL MOVE(BLANKS, LINE,-NWL) CALL CHOP(CIT,JC, LINE,0,NC) </pre>	<pre> PRNCIT 2 PRNCIT 3 PRNCIT 4 PRNCIT 5 PRNCIT 6 PRNCIT 7 PRNCIT 8 PRNCIT 9 PRNCIT10 PRNCIT11 PRNCIT12 PRNCIT13 PRNCIT14 PRNCIT15 PRNCIT16 PRNCIT17 PRNCIT18 PRNCIT19 PRNCIT20 PRNCIT21 PRNCIT22 PRNCIT23 PRNCIT24 PRNCIT25 PRNCIT26 PRNCIT27 PRNCIT28 PRNCIT29 PRNCIT30 PRNCIT31 PRNCIT32 PRNCIT33 PRNCIT34 PRNCIT35 PRNCIT36 PRNCIT37 PRNCIT38 PRNCIT39 PRNCIT40 PRNCIT41 PRNCIT42 PRNCIT43 PRNCIT44 PRNCIT45 PRNCIT46 PRNCIT47 PRNCIT48 PRNCIT49 PRNCIT50 PRNCIT51 PRNCIT52 PRNCIT53 PRNCIT54 PRNCIT55 PRNCIT56 PRNCIT57 PRNCIT58 </pre>

```

      CALL PAGER(1)
      IF (.NOT.TOP) GO TO 24
      IF (JCIT.GF.1) PRINT 1000,LCIT,FTAG,LINE,JCIT
1000  FORMAT(1X,I4,2X,A4,2X,13A8,A5,I6)
      IF (JCIT.LT.1) PRINT 2000,LCIT,FTAG,LINE
2000  FORMAT(1X,I4,2X,A4,2X,13A8,A5)
      GO TO 26
      24 PRINT 3000,FTAG,LINE
3000  FORMAT(7X,A4,2X,13A8,A5)
      26 TOP =.FALSE.
      FTAG=BLANKS
      JC=JC+NC
      LEFT=LEFT-NC
      IF (LEFT.GF.1) GO TO 10
39  CONTINUE
      RETURN
      ND

```

```

PRNCIT59
PRNCIT60
PRNCIT61
PRNCIT62
PRNCIT63
PRNCIT64
PRNCIT65
PRNCIT66
PRNCIT67
PRNCIT68
PRNCIT69
PRNCIT70
PRNCIT71
PRNCIT72
PRNCIT73
PRNCIT74
PRNCIT75

```

SUBROUTINE PRNDIC(KON)	PRNDIC 2
C PRINTS EITHER THE TRIVIAL OR THE VOCABULARY PORTION OF THE DICTIONARY,	PRNDIC 3
C FOR VOCAB,	PRNDIC 4
C KON=0, PRINT THE TRIVIAL WORDS, IN ALPHETICAL ORDER,	PRNDIC 5
C KON=1, PRINT THE VOCABULARY ENTRIES, IN ALPHARETICAL ORDER.	PRNDIC 6
C KON=2, PHINT ALL ENTRIES, IN ORDER OF OCCURRANCE IN DICT,	PRNDIC 7
COMMON /CORE/NCHW,BLANKS,BLANK	PRNDIC 8
INTEGER BLANKS,BLANK	PRNDIC 9
COMMON /FAM/SLASH,STAR,COMMA,ZERO,MXCHAR,MXCDIS,MXWDES,MXCHW,	PRNDIC10
MXFREE,MXWFRE	PRNDIC11
INTEGER SLASH,STAR,COMMA,ZERO	PRNDIC12
COMMON NDICT,NVOC,NTRIV,MXHASH,MXENT,MXDICT,WORD(5),	PRNDIC13
DICT(6750),LDICT(6501),HASH(9511)	RJO 14
COMMON /FREQY/ NFREQY(4,6500)	HJO 23
INTEGER MLINE(4,3)	RJO 24
INTEGER DICT,HASH,WORD,KRANK(6500)	PRNDIC15
INTEGER P(3)	JRT
EQUIVALENCE(KRANK,HASH)	PRNDIC16
INTEGER LINE(5,3)	PRNDIC17
C LINE CONTAINS THREE DICTIONARY ENTRIES, FOR PRINTING ON THE CURRENT	PRNDIC18
C LINE, THE FIRST DIMENSION OF LINE IS MXWDES,	PRNDIC19
NLINES=50	PRNDIC20
IF (KON.NE. 2) NLINES = 25	RJO 25
KN=KON+1	PRNDIC21
GO TO (10,20,30),KN	PRNDIC22
10 LOUT=0	PRNDIC23
LEFT=NTRIV	PRNDIC24
GO TO 40	PRNDIC25
20 LOUT=NTRIV	PRNDIC26
LEFT=NVOC	PRNDIC27
GO TO 40	PRNDIC28
30 LOUT=0	PRNDIC29
LEFT=NDICT	PRNDIC30
40 MOST=LOUT+LEFT	PRNDIC31
50 IF(LEFT.LE.0) GO TO 200	PRNDIC32
CALL PAGER(0)	PRNDIC33
CALL PAGER(2)	PRNDIC34
GO TO (60,70,80),KN	PRNDIC35
60 PRINT 1000	PRNDIC36
1000 FORMAT(41X,25HT R I V I A L W O R D S/1X)	PRNDIC37
GO TO 100	PRNDIC38
70 PRINT 1001	PRNDIC39
1001 FORMAT(44X,19HV O C A B U L A R Y/1X)	PRNDIC40
GO TO 100	PRNDIC41
80 CONTINUE	PRNDIC42
PRINT 1002	PRNDIC43
1002 FORMAT(44X,19HD I C T I O N A R Y/1X)	PRNDIC44
100 NL=MIND(NLINES,LEFT)	PRNDIC45
NJ=3	PRNDIC46
CALL PAGER(NL)	PRNDIC47
DO 130 L=1,NL	PRNDIC48
DO 145 IY= 1,3	JRT
DO 145 IZ= 1,4	JRT
145 MLINE(IZ,IY) = 0	JRT
CALL MOVE(BLANKS,LINE,-NJ*MXWDES)	PRNDIC49
DO 110 J=1,NJ	PRNDIC50
K=LOUT+(J-1)*NLINES+L	PRNDIC51

```

      IF(K,GT,MOST) GO TO 120
      KR=K
      IF(KON,NE,2) KR=KRANK(K)
C---STORE FREQUENCY COUNT AND CITATION NUMBERS FOR 3 REFERENCES
C---ADDED IN JULY 1975 BY RJ OLSON
      DO 105 LL = 1,4
      MLINE(LL,J) = NFREQY(LL,KR)
105  CONTINUE
      LD=IABS(LDICT(KR))
      ND=IABS(LDICT(KR+1))-LD
      CALL CHOP(DICT,LD,LINE(1,J),0,ND)
      IF(KON,NE,2.OR.ISIGN(1,LDICT(KR)).GT,0) GO TO 110
      CALL CHAR(STAR,LINE(1,J),MIN0(ND+1,MXWDES))
110  CONTINUE
120  PRINT 2000,((LINE(I,J),I=1,MXWDES),J=1,NJ)
2000  FORMAT(2X,3(5A8,2X))
      IF (KON ,EQ. 2) GO TO 140
C---ONLY PRINT ITEMS WITH VALUES
      DO 135 I = 1,3
      P(I) = "((((((( ))) ) ) ) )"
      IF ( MLINE(1,I).LE. 0) P(I) = " "
135  CONTINUE
      PRINT 3000,MLINE,P,P
C      SUPPRESS LEFT AND RIGHT PARENS, IF NUMBERS ARE 0
C      R1 SPEC IN THE FORMAT CAUSES THE RIGHT HAND CHAR TO PRINT
C      A1 SPEC IN THE FORMAT CAUSES THE LEFT- HAND CHAR TO PRINT
3000  FORMAT(2X, 3 ( 15X,I4.0,1X,3I5.0,5X), .
      + T22,A1,T62,A1,T102,A1 , T38,R1 , T78,R1 , T118,R1 )
140  CONTINUE
130  CONTINUE
      LOUT=LOUT+NJ*NL
      LEFT=LEFT-NJ*NL
      GO TO 50
200  CALL PAGER(C)
      RETURN
      tND

```

```

PRNDIC52
PRNDIC53
PRNDIC54
RJO 26
RJO 27
RJO 28
RJO 29
RJO 30
PRNDIC55
PRNDIC56
PRNDIC57
PRNDIC58
PRNDIC59
PRNDIC60
PRNDIC61
PRNDIC62
RJO 31
RJO 32
RJO 34
JRT
JRT
RJO 37
JRT
JRT
JRT
JRT
JRT
RJO 40
PRNDIC63
PRNDIC64
PRNDIC65
PRNDIC66
PRNDIC67
PRNDIC68
PRNDIC69

```

SUBROUTINE PUNDIC	PUNDIC 2
C PUNCHES OUT THE DICTIONARY, IN ALPHABETICAL ORDER, FOR VOCAB.	PUNDIC 3
COMMON /CORE/NCHW,BLANKS,BLANK	PUNDIC 4
INTEGER BLANKS,BLANK	PUNDIC 5
COMMON /FAM/SLASH,STAR,COMMA,ZERO,MXCHAR,MXCODES,MXWDES,MXCHW,	PUNDIC 6
MXFREE,MXWFRE	PUNDIC 7
INTEGER SLASH,STAR,COMMA,ZERO	PUNDIC 8
COMMON NDICT,NVOC,NTRIV,MXHASH,MXENT,MXDICT,WORD(5),	PUNDIC 9
DICT(6750),LDICT(6501),HASH(9511)	PUNDIC10
INTEGER DICT,HASH,WORD,KRANK(6500)	PUNDIC11
EQUIVALENCE(KRANK,HASH)	PUNDIC12
INTEGER KARD(10)	PUNDIC13
C KARD CONTAINS THE CURRENT CARD IMAGE BEING SET UP F3K PUNCHING. ITS	PUNDIC14
C DIMENSION IS NWC.	PUNDIC15
NWC=(80+1CHW-1)/NCHW	PUNDIC16
LC=0	PUNDIC17
C MAJOR LOOP.	PUNDIC18
DO 60 K=1,NDICT	PUNDIC19
KH=KRANK(K)	PUNDIC20
LD=IABS(LDICT(KH))	PUNDIC21
ND=IABS(LDICT(KH+1))-LD	PUNDIC22
C PUNCH DICTIONARY ENTRY.	PUNDIC23
10 IF(LC.LT.80) GO TO 20	PUNDIC24
PUNCH 1000, (KARD(J),J=1,NWC)	PUNDIC25
1000 FORMAT(10A8)	PUNDIC26
LC=0	PUNDIC27
20 LC=LC+1	PUNDIC28
LD=LD+1	PUNDIC29
ND=ND-1	PUNDIC30
KH=KHAR(DICT,LD)	PUNDIC31
CALL CHAR(KH,KARD,LC)	PUNDIC32
IF(ND.GT.0) GO TO 10	PUNDIC33
C INSERT COMMA OR ASTERISK.	PUNDIC34
IF(K.EQ.NDICT) GO TO 60	PUNDIC35
KH=COMMA	PUNDIC36
IF(K.LE.NTRIV) KH=STAR	PUNDIC37
IF(LC.LT.80) GO TO 30	PUNDIC38
PUNCH 1000, (KARD(J),J=1,NWC)	PUNDIC39
LC=0	PUNDIC40
30 LC=LC+1	PUNDIC41
CALL CHAR(KH,KARD,LC)	PUNDIC42
60 CONTINUE	PUNDIC43
C PUNCH OUTSTANDING CARD.	PUNDIC44
CALL CHOP(BLANKS,0,KARD,LC,-(80-LC))	PUNDIC45
PUNCH 1000, (KARD(J),J=1,NWC)	PUNDIC46
RETURN	PUNDIC47
END	PUNDIC48
C-C-C-SUBROUTINE REWIND(MT)	REWIND 2
C REWINDS TAPE MT.	REWIND 3
CALL BUTLER	REWIND 4
REWIND MT	REWIND 5
C-C-C CALL KORAN(6HREWIND,MT,1)	REWIND 6
CALL KORAN(6HREWAND,MT,1)	REWIND 6
RETURN	REWIND 7
END	REWIND 8

SUBROUTINE SCAN(KD,KODE,X,LX,NX,NSKIP,KH)	SCAN	2
C SCANS A CHARACTER STRING X FROM POSITION LX+1 THROUGH LX+NX LOOKING	SCAN	3
C FOR ONE WHICH MATCHES ANY ONE OF A GIVEN SET OF CHARACTERS.	SCAN	4
C THE SET OF CHARACTERS TO SEARCH FOR IS GIVEN BY THE 64-CELL ARRAY	SCAN	5
C KODE. THE J-TH CELL OF KODE REPRESENTS THE CHARACTER WHOSE INTERNAL	SCAN	6
C CODE IS J-1. THUS ALL OF THE POSSIBLE 64 CHARACTERS MAY BE	SCAN	7
C REPRESENTED. IF KODE(J)=KD, CHARACTER J-1 IS ONE OF THE SEARCHED-FOR	SCAN	8
C CHARACTERS. IF KODE(J) IS ANY OTHER VALUE, CHAR J-1 IS NOT SEARCHED	SCAN	9
C FOR.	SCAN	10
C THE RESULT IS INDICATED BY NSKIP AND KH. NSKIP IS THE NUMBER OF CHARS	SCAN	11
C NOT IN THE GIVEN SET THAT WERE SKIPPED OVER BEFORE ONE IN THE SET WAS	SCAN	12
C FOUND. IF NONE OF THE NX CHARACTERS WERE IN THE SET, NSKIP = NX.	SCAN	13
C KH IS THE CHARACTER CODE OF THE ONE FOUND, IF ANY.	SCAN	14
C IN CASE NX IS LESS THAN 1, SETS NSKIP=0 AND RETURNS.	SCAN	15
C FOR SPEED, IT IS ADVISABLE TO HAND CODE THIS SUBROUTINE.	SCAN	16
INTEGER KODE(64), X(2)	SCAN	17
IF(NX .LT. 1) GO TO 40	SCAN	18
JA = LX+1	SCAN	19
JB=LX+NX	SCAN	20
DO 10 J=JA,JB	SCAN	21
KH=KHAR(X,J)	SCAN	22
IF(KODE(KH+1),EQ,KD) GO TO 30	SCAN	23
10 CONTINUE	SCAN	24
SKIP=NX	SCAN	25
RETURN	SCAN	26
30 NSKIP=J-LX-1	SCAN	27
RETURN	SCAN	28
40 NSKIP=0	SCAN	29
RETURN	SCAN	30
END	SCAN	31

SUBROUTINE SCAT(X,LX,NX,DICT,LDICT,NDICT,LASH,NLASH,JDICT,JLASH)	SLAT	2
C GIVEN A WORD WHICH IS A SUBSTRING OF X CONSISTING OF NX CHARACTERS	SCAT	3
C WITH LX POINTING TO ITS POSITION IN X, AND GIVEN A DICTIONARY DICT	SCAT	4
C HAVING NDICT ENTRIES WITH LDICT(J) POINTING TO ITS J-TH ENTRY (STORED	SCAT	5
C AS A SUBSTRING OF DICT), AND GIVEN A HASH CODE TABLE LASH (SEE ROBERT	SCAT	6
C MORRIS, SCATTER STORAGE TECHNIQUES, COMM. A.C.M., JAN. 1968) WHOSE	SCAT	7
C LENGTH IS NLASH, WE DETERMINE WHETHER THE GIVEN WORD IS ALREADY IN	SCAT	8
C DICT. IF SO, JDICT IS RETURNED SHOWING WHICH ONE IT IS. IF NOT, JDICT	SCAT	9
C IS RETURNED AS NDICT+1. IN EITHER EVENT, JLASH IS RETURNED AS ITS HASH	SCAT	10
C CODE, I.E., ITS POSITION IN THE ARRAY LASH,	SCAT	11
C X, DICT, AND LASH ARE NOT CHANGED,	SCAT	12
C SOME OF THE LDICT(J) VALUES MAY BE NEGATIVE, INDICATING THAT THE J-TH	SCAT	13
C ENTRY IS A TRIVIAL WORD, THUS, IABS(LDICT(J)) IS ACTUALLY THE	SCAT	14
C POINTER TO THE J-TH ENTRY,	SCAT	15
COMMON /FAM/SLASH,STAR,COMMA,ZERO,MXCHAR,MXCDER,MXWDER,MXCHW,	SCAT	16
MXFREE,MXWFER	SCAT	17
INTEGER SLASH,STAR,COMMA,ZERO	SCAT	18
COMMON /CORE/NCHW,BLANKS,BLANK	SCAT	19
INTEGER BLANKS,BLANK	SCAT	20
INTEGER X(2),DICT(2),LDICT(2),LASH(2),WORD(5)	SCAT	21
KUT(N)=(N+NCHW-1)/NCHW	SCAT	22
C GET WORD FROM X,	SLAT	23
CALL MOVE(BLANKS,WORD,-MXWDER)	SCAT	24
CALL CHOP(X,LX,WORD,0,MINO(NX,MXCDER))	SCAT	25
C COMPUTE HASH CODE	SCAT	26
CALL HASH(WORD,KUT(NX),NLASH,LH)	SCAT	27
CALL KORAN(4HHASH,LH,1)	SCAT	28
INC=NLASH/2	SCAT	29
GO TO 15	SCAT	30
C QUADRATIC SEARCH,	SCAT	31
10 LH=LH+INC	SCAT	32
IF(LH,LE,0) LH=LH+NLASH	SCAT	33
15 JD=LASH(LH)	SCAT	34
IF(JD,EQ,0) GO TO 30	SCAT	35
LD=IABS(LDICT(JD))	SCAT	36
ND=IABS(LDICT(JD+1))-LD	SCAT	37
CALL SIMILE(1,DICT,LD,ND,WORD,0,NX,LIKE)	SCAT	38
IF(LIKE,EQ,1) GO TO 40	SCAT	39
INC=INC+1	SCAT	40
IF(INC,GT,0) GO TO 10	SCAT	41
C QUADRATIC SEARCH FAILS, SEARCH ENTIRE TABLE,	SCAT	42
DO 20 LH=1,NLASH	SCAT	43
JD=LASH(LH)	SCAT	44
IF(JD,EQ,0) GO TO 30	SCAT	45
LD=IABS(LDICT(JD))	SCAT	46
ND=IABS(LDICT(JD+1))-LD	SCAT	47
CALL SIMILE(1,DICT,LD,ND,WORD,0,NX,LIKE)	SCAT	48
IF(LIKE,EQ,1) GO TO 40	SCAT	49
20 CONTINUE	SCAT	50
STOP	SCAT	51
C FOUND AN EMPTY SLOT,	SCAT	52
30 JDICT=NDICT+1	SCAT	53
JLASH=LH	SCAT	54
RETURN	SCAT	55
C LOCATED WORD IN DICT,	SCAT	56
40 JDICT=JD	SCAT	57
JLASH=LH	SCAT	58
RETURN	SCAT	59
END	SCAT	60

SUBROUTINE SCOUT(LR,MD,KH,X,LX,NX,NFOUND)	SCOUT 2
C SEARCHES A SUBSTRING OF A CHAP STRING, BEGINNING AT A GIVEN POINT,	SCOUT 3
C LOOKING FOR A CONSECUTIVE STRING OF CHARACTERS THAT MATCH A GIVEN	SCOUT 4
C CHARACTER, OR LOOKS FOR A CONSECUTIVE STRING ALL DIFFERENT FROM THE	SCOUT 5
C CHARACTER.	SCOUT 6
C KH IS THE GIVEN CHARACTER (THE INTEGER VALUE OF ITS CODE).	SCOUT 7
C X IS THE CHARACTER ARRAY TO BE SEARCHED. X IS PACKED 1 CHW CHARACTERS	SCOUT 8
C PER WORD.	SCOUT 9
C LX POINTS TO THE START OF THE SUBSTRING. I.E., IT STARTS AT POSITION	SCOUT 10
C LX+1 IN X.	SCOUT 11
C NX IS THE NO. OF CHARS IN THE SUBSTRING.	SCOUT 12
C NFOUND IS THE NUMBER OF CONSECUTIVE CHARACTERS FOUND THAT SATISFY THE	SCOUT 13
C REQUIREMENT. (RETURNED)	SCOUT 14
C THE TYPE OF SEARCH IS DETERMINED BY LR AND MD.,,	SCOUT 15
C LR=0, SEARCH LEFT TO RIGHT, BEGINNING AT LX+1.	SCOUT 16
C LR=1, SEARCH RIGHT TO LEFT, BEGINNING AT LX + NX.	SCOUT 17
C MD=0, SEARCH FOR CONSECUTIVE MATCHING CHARACTERS.	SCOUT 18
C MD=1, SEARCH FOR CONSECUTIVE DIFFERENT CHARACTERS.	SCOUT 19
C NFOUND MAY OF COURSE BE 0, EITHER BECAUSE THE VERY FIRST CHARACTER	SCOUT 20
C FAILS THE CRITERION OR BECAUSE NX IS LESS THAN 1, INDICATING AN EMPTY	SCOUT 21
C SUBSTRING OF X.	SCOUT 22
C FGH SPEED, IT IS ADVISABLE TO HAND CODE THIS SUBROUTINE.	SCOUT 23
INTEGER X(2)	SCOUT 24
IF(NX.GT.0) GO TO 10	SCOUT 25
NFOUND=0	SCOUT 26
RETURN	SCOUT 27
10 IF(LR.NE.0) GO TO 60	SCOUT 25
JA=LX+1	SCOUT 23
JB=LX+NX	SCOUT 30
C LEFT TO RIGHT SCAN.	SCOUT 31
IF(MD.NE.0) GO TO 30	SCOUT 32
C MATCHING.	SCOUT 33
DO 20 J=JA,JB	SCOUT 34
IF(KHAR(X,J).NE.KH) GO TO 55	SCOUT 35
20 CONTINUE	SCOUT 36
GO TO 50	SCOUT 37
C DIFFERENT.	SCOUT 38
30 DO 40 J=JA,JB	SCOUT 39
IF(KHAR(X,J).EQ.KH) GO TO 55	SCOUT 40
40 CONTINUE	SCOUT 41
50 J=JB+1	SCOUT 42
55 NFOUND=J-JA	SCOUT 43
RETURN	SCOUT 44
C RIGHT TO LEFT SCAN.	SCOUT 45
60 L=LX+NX	SCOUT 46
IF(MD.NE.0) GO TO 80	SCOUT 47
C MATCHING.	SCOUT 48
DO 70 J=1,NX	SCOUT 49
IF(KHAR(X,L).NE.KH) GO TO 110	SCOUT 50
70 L=L-1	SCOUT 51
GO TO 100	SCOUT 42
C DIFFERENT.	SCOUT 53
60 DO 90 J=1,NX	SCOUT 54
IF(KHAR(X,L).EQ.KH) GO TO 110	SCOUT 55
90 L=L-1	SCOUT 56
C RETURN.	SCOUT 57
100 J=NX+1	SCOUT 58
110 NFOUND=J-1	SCOUT 59
RETURN	SCOUT 60
END	SCOUT 61

SUBROUTINE SETCOM	SETCOM 2
C SETS CONSTANTS IN THOSE COMMON BLOCKS USED THROUGHOUT FAMULUS.	SETCOM 3
C	SETCOM 4
COMMON /CORE/NCHW,BLANKS,BLANK	SETCOM 5
INTEGER BLANKS,BLANK	SETCOM 6
C	SETCOM 7
COMMON /DSCRIP/IDSCRIP	SETCOM 8
C	SETCOM 9
COMMON /ELIN/INTAPE	SETCOM 10
C	SETCOM 11
COMMON /FAM/SLASH,STAR,COMMA,ZERO,MXCHAR,MXCDDES,MXWDES,MXCHW,	SETCOM 12
,MXFREE,MXWFHE	SETCOM 13
INTEGER SLASH,STAR,COMMA,ZERO	SETCOM 14
C	SETCOM 15
COMMON /INPT/CARD(10),KOLUM,HOLD,ENDCAR,BLACAR,CONCAR,ERRIN,NWC,	SETCOM 16
,MXFL,MAXCAR,NCARIN,KHR,KIN,LIT,LITX,SYM(5),NUMB,NCSYM,CATLOG,	SETCOM 17
,KARA,KARB	SETCOM 18
INTEGER CARD,SYM	SETCOM 19
LOGICAL HOLD,ENDCAR,BLACAR,CONCAR,ERRIN,CATLOG	SETCOM 20
C	SETCOM 21
COMMON /TEST/ITEST	SETCOM 22
C	SETCOM 23
INTEGER BLS	SETCOM 24
DATA BLS/1H /	SETCOM 25
KUT(N,M)=(N+M-1)/M	SETCOM 26
C MACHINE-DEPENDENT CONSTANTS.	SETCOM 27
NCHW=8	SETCOM 28
MXCHW=10	SETCOM 29
NWC=KUT(80,NCHW)	SETCOM 30
IDSCRIP=0	SETCOM 31
INTAPE=50	SETCOM 32
C FAMULUS CONSTANTS,	SETCOM 33
MXCHAR=4000	SETCOM 34
MXCDDES=40	SETCOM 35
MXWDES=KUT(MXCDDES,NCHW)	SETCOM 36
MXFREE=100	SETCOM 37
MXWFRE=KUT(MXFREE,NCHW)	SETCOM 38
MAXCAR=65	SETCOM 39
MXFL=4	SETCOM 40
CATLOG=,FALSE,	SETCOM 41
KARA=6	SETCOM 42
KARB=80	SETCOM 43
C CHARACTER CONSTANTS.	SETCOM 44
SLASH=KHAR(1H/,1)	SETCOM 45
STAR=KHAR(1H*,1)	SETCOM 46
COMMA=KHAR(1H,,1)	SETCOM 47
ZERO=KHAR(1H0,1)	SETCOM 48
BLANK=KHAR(1H ,1)	SETCOM 49
BLANKS=BLS	SETCOM 50
C INITIALIZE.	SETCOM 51
NCARIN=0	SETCOM 52
ENDCAR=,FALSE,	SETCOM 53
ERRIN=,FALSE,	SETCOM 54
ITEST=0	SETCOM 55
RETURN	SETCOM 56
C	SETCOM 57
C /CORE/	SETCOM 58

C		SETCOM59
C NCHW	NUMBER OF CHARACTERS IN A PACKED CELL, IN FAMULUS, THIS	SETCOM60
C	IS DETERMINED BY SUBROUTINE CHAR AND FUNCTION KHPR AND	SETCOM61
C	IS CHOSEN FOR SPEED IN THOSE TWO ROUTINES, IT IS	SETCOM62
C	THEREFORE NOT NECESSARILY EQUAL TO MXCHW, (8,4)	SETCOM63
C BLANKS	CELL FULL OF BLANK CHARACTERS,	SETCOM64
C BLANK	INTERNAL CHARACTER CODE FOR A BLANK CHARACTER, (451)	SETCOM65
C		SETCOM66
C /DSCR/		SETCOM67
C		SETCOM68
C IDSCR	NORMALLY IDSCR=0, BUT POLISH SETS IDSCR=1 TO TELL	SETCOM69
C	SYMBOL SUBROUTINE TO COLLECT DESCRIPTOR SYMBOLS USING	SETCOM70
C	MORE LIBERAL RULES THAN USUAL, THAT IS, THEY MAY BEGIN	SETCOM71
C	WITH A DIGIT AND MAY INCLUDE ANY CHARACTERS EXCEPT	SETCOM72
C	PARENTHESES, COMMA, AND THE THREE LOGICAL OPERATORS.	SETCOM73
C		SETCOM74
C /EDIN/		SETCOM75
C		SETCOM76
C INTAPE	INPUT UNIT CONTAINING CARD IMAGE INPUT TO EDIT, NORMALLY	SETCOM77
C	THE SYSTEM INPUT, THE ONLY TIME IT IS SOMETHING ELSE	SETCOM78
C	IS WHILE READING THE BODY OF A /CITATIONS ON TAPE/ STMT.	SETCOM79
C		SETCOM80
C /FAM/		SETCOM81
C		SETCOM82
C SLASH	INTERNAL CHARACTER CODE FOR A SLASH, (40,)	SETCOM83
C STAR	INTERNAL CHARACTER CODE FOR AN ASTERISK, (39,)	SETCOM84
C COMMA	INTERNAL CHARACTER CODE FOR A COMMA, (46,)	SETCOM85
C ZERO	INTERNAL CHARACTER CODE FOR A ZERO, (27,)	SETCOM86
C MXCHAR	MAX. NO. OF CHARS. IN A CITATION, (4000)	SETCOM87
C MXCDES	MAX. NO. OF CHARS. IN A DESCRIPTOR, (40)	SETCOM88
C MXWDES	MAX. NO. OF CELLS IN A DESCRIPTOR, (5,10)	SETCOM89
C MXCHW	MAX. NO. OF CHARS. A CELL HAS ROOM FOR, (10,4)	SETCOM90
C MXFREE	MAX. NO. OF CHARS. IN A FREE-FORM STATEMENT, SUCH AS	SETCOM91
C	AY /ID/ STATEMENT, (100)	SETCOM92
C MXWFRE	SAME AS MXFREE, ONLY IT IS THE NO. OF CELLS, (13,26)	SETCOM93
C		SETCOM94
C /FIEL/		SETCOM95
C		SETCOM96
C FIELD(10)	FIELD LABELS.	SETCOM97
C IFIELD(10)	A PERMUTATION OF 1,2,...,NFIELD INDICATING AN ORDERING	SETCOM98
C	OF THE FIELDS. HOWEVER, THE FIELDS ARE ALWAYS STORED IN	SETCOM99
C	THEIR ORIGINAL ORDER WHILE IN CORE OR ON TAPE, THEY ARE	SETCOM100
C	REORDERED ONLY ON PRINTOUT.	SETCOM101
C NFIELD	NUMBER OF FIELDS DEFINED BY THE USER.	SETCOM102
C KDESCR	DESCRIPTOR FIELD, IF NONE, KDESCR=0.	SETCOM103
C		SETCOM104
C /INPT/		SETCOM105
C		SETCOM106
C CARD(NWC)	ARRAY CONTAINING THE CURRENT INPUT CARD,	SETCOM107
C KOLUM	COLUMN NUMBER OF LAST CHARACTER TAKEN FROM CURRENT	SETCOM108
C	INPUT CAQC,	SETCOM109
C HOLD	TRUE IFF THE CURRENT CARD IS TO BE PRINTED BUT HAS NOT	SETCOM110
C	YET BEEN PRINTED,	SETCOM111
C ENDCAR	TRUE IFF THERE ARE NO MORE CARDS IN INPUT,	SETCOM112
C BLACAR	TRUE IFF THE CURRENT INPUT CARD IS ENTIRELY BLANK,	SETCOM113
C CONCAH	TRUE IFF THE CURRENT INPUT CARD IS THE BEGINNING OF A	SETCOM114
C	STATEMENT (SLASH IN COLUMN 1),	SETCOM115

C ERRIN	TRUE IFF AN ERROR HAS BEEN DETECTED IN THE INPUT CARDS.	SETC0116
C KHR	INTERNAL CODE OF THE CURRENT INPUT CARD CHARACTER,	SETC0117
C	PRODUCED BY FUNCTION KHAR.	SETC0118
C KIN	TYPE OF KHR (PRODUCED BY FUNCTION IDEAL).	SETC0119
C LIT	INDICATES THE TYPE OF THE CURRENT SYMBOL. SEE COMMENTS	SETC0120
C	IN SUBROUTINE SYMBOL.	SETC0121
C LITX	SAME AS LIT BUT REFERS TO THE PREVIOUS SYMBOL.	SETC0122
C SYM(MXWDES)	CURRENT SYMBOL (BLANK IF NUMERIC).	SETC0123
C NUMB	NUMERIC VALUE OF CURRENT SYMBOL, IF INTEGER.	SETC0124
C MAXCAR	MAXIMUM NUMBER OF CARDS IN A CITATION. (65)	SETC0125
C NCARIN	NUMBER OF CARDS READ SO FAR.	SETC0126
C NCSYM	NUMBER OF CHARACTERS IN THE CURRENT SYMBOL.	SETC0127
C NWC	NUMBER OF CELLS IN WHICH A CARD IMAGE IS STORED WHEN	SETC0128
C	PACKED NCHW CHARACTERS PER CELL. (10,20)	SETC0129
C KARA	COLUMN WHERE TEXT BEGINS ON A CITATION INPUT CARD. (6)	SETC0130
C KARB	COLUMN WHERE TEXT ENDS ON A CITATION INPUT CARD. (80)	SETC0131
C CATLOG	TRUE IFF THE CITATION CARD INPUT IN EDIT IS IN THE	SETC0132
C	(OBSOLETE) CATALOG FORMAT.	SETC0133
C MXFL	MAXIMUM NUMBER OF CHARACTERS IN A FIELD LABEL. (4)	SETC0134
C		SETC0135
C /TEST/		SETC0136
C		SETC0137
C ITEST	ITEST,NE,0 MEANS DEBUG PRINTOUT IS WANTED.	SETC0138
END		SETC0139

SUBROUTINE SETSYM	SETSYM 2
C INITIALIZES FOR SYMBOL COLLECTION AT THE START OF INPUT OF A	SETSYM 3
C STATEMENT.	SETSYM 4
C SETS CURRENT SYMBOL INDICATOR (LIT) TO INDICATE END OF A STATEMENT,	SETSYM 5
I.E. THE PREVIOUS STATEMENT.	SETSYM 6
SETS CARD COLUMN POINTER (KOLUM) AT 0.	SETSYM 7
C IF THE CURRENT CARD IS NOT A CONTROL CARD, AN ERROR FLAG IS SET AND	SETSYM 8
C A MESSAGE IS PRINTED. NO OTHER ACTION IS TAKEN.	SETSYM 9
COMMON /INPT/CARD(10),KOLUM,HOLD,ENDCAR,PLACAR,CONCAR,ERRIN,NWC,	SETSYM10
MXFL,MAXCAR,NCARIN,KHR,KIN,LIT,LITX,SYM(5),NUMB,NCSYM,CATLOG,	SETSYM11
KARA,KARB	SETSYM12
INTEGER CARD,SYM	SETSYM13
LOGICAL HOLD,ENDCAR,PLACAR,CONCAR,ERRIN,CATLOG	SETSYM14
CALL BUTLER	SETSYM15
IF(NCARIN.LE.0) GO TO 15	SETSYM16
10 IF(ENDCAR) GO TO 30	SETSYM17
IF(PLACAR) GO TO 15	SETSYM18
IF(KOLUM.GT.1) GO TO 15	SETSYM19
IF(CONCAR) GO TO 20	SETSYM20
15 CALL INCARD	SETSYM21
GO TO 10	SETSYM22
20 KOLUM=0	SETSYM23
30 LITX=12	SETSYM24
LIT=12	SETSYM25
KIN=1	SETSYM26
C CALL SPOT(LIT,LITX,KHR,KIN,4HIIAI,24HLIT LITX KHR KIN)	SETSYM27
C CALL SPOT(NUMB,NCSYM,NCARIN,KOLUM,4HIIII,	SETSYM28
, 24HNUMB NCSYM NCARINKOLUM)	SETSYM29
C CALL SPOT(ERRIN,ENDCAR,CONCAR,HOLD,4HLLLL,	SETSYM30
C 24HERRIN ENDCARCONCARHOLD)	SETSYM31
RETURN	SETSYM32
END	SETSYM33

SUBROUTINE SHELLI(X,IRANK,N)	SHELL
C-----SHELL CHANGED TO SHELLI BY JOHN R. TADLOCK NOV. 1975,	JRT
C-----THIS WAS DONE TO ELIMINATE A LDSET CARD IN USING PROGRAM INDEX	JRT
C-----FOR AN EXPLANATION OF HOW THIS SUBROUTINE WORKS REFER TO	JRT
C-----COMMENTS ON SUBROUTINE SHELLS,	JRT
C-----THERE ARE 3 VERSIONS OF SHELL AND ASCEND IN THE FAMULUS LIBRARY	JRT
C-----INDEX CALLS SUBROUTINE SHELLI WHICH CALLS FUNCTION ASCEND	JRT
C-----SORT CALLS SUBROUTINE SHELLS WHICH CALLS FUNCTION ASCENDS	JRT
C-----VOCAB CALLS SUBROUTINE SHELLV WHICH CALLS FUNCTION ASCENV	JRT
C	SHELL 44
C THIS PROGRAM WAS WRITTEN IN ASA STANDARD FORTRAN BY ROBERT M. RUSSELL,	SHELL 45
C MAY 1967,	SHELL 46
LOGICAL ASCEND	SHELL 47
C---- LOGICAL ASCEND	SHELL 47
INTEGER IRANK(N)	SHELL 48
IF(N,LE,0) GO TO 100	SHELL 49
DO 10 J=1,N	SHELL 50
10 IRANK(J)=J	SHELL 51
IF(N,EQ,1) GO TO 100	SHELL 52
M=1	SHELL 53
20 M=M+M	SHELL 54
IF(M,LT,N) GO TO 20	SHELL 55
M=MAX0(1,M/2-1)	SHELL 56
30 I=1	SHELL 57
IM=I+M	SHELL 58
40 J=I	SHELL 59
JM=IM	SHELL 60
IHOLD=IRANK(IM)	SHELL 61
50 JR=IRANK(J)	SHELL 62
IF(ASCEND(X,JR,IHOLD)) GO TO 60	SHELL 63
C---- IF(ASCEND(X,JR,IHOLD)) GO TO 60	SHELL 63
IRANK(JM)=IRANK(J)	SHELL 64
JM=J	SHELL 65
J=J-M	SHELL 66
IF(J,GT,0) GO TO 50	SHELL 67
60 IRANK(JM)=IHOLD	SHELL 66
I=I+1	SHELL 69
IM=I+M	SHELL 70
IF(IM,LE,N) GO TO 40	SHELL 71
M=M/2	SHELL 72
IF(M,GT,0) GO TO 30	SHELL 73
100 RETURN	SHELL 74
END	SHELL 75

```

SUBROUTINE SHELLS(X,IRANK,N)
C-----
C-----SUBROUTINE SHELL CHANGED TO SHELLS BY JOHN R. TA LOCK NOV. 1975.
C-----THERE ARE 3 VERSIONS OF SHELL AND ASCEND IN THE FAMULUS LIBRARY
C-----INDEX CALLS SUBROUTINE SHELLI WHICH CALLS FUNCTION ASCEND
C-----SORT CALLS SUBROUTINE SHELLS WHICH CALLS FUNCTION ASCEND
C-----VOCAB CALLS SUBROUTINE SHELLV WHICH CALLS FUNCTION ASCENV
C DOES AN INTERNAL SORT ON N KEYS, LOCATED AT X, SORTS INTO ASCENDING
C ORDER, USING SHELL'S METHOD.
C THIS SUBROUTINE IS A GENERAL APPLICATION OF SHELL'S METHOD ALLOWING
C THE PROGRAMMER TO SORT KEYS OF ANY STRUCTURE. THIS IS ACCOMPLISHED BY
C REQUIRING THE PROGRAMMER TO SUPPLY A FUNCTION, ASCEND, WHICH COMPARES
C ANY TWO KEYS FOR PRECEDENCE, ASCEND IS DEFINED AS FOLLOWS,
C ASCEND(X,I,J) IS TRUE IF THE I-TH KEY PRECEDES THE J-TH KEY, FALSE
C OTHERWISE.
C HOWEVER, IT IS COMMON THAT THE PROGRAMMER SIMPLY WANTS TO SORT AN
C ARRAY OF NUMBERS, FOR WHICH PURPOSE TWO VARIATIONS OF SHELL ARE
C AVAILABLE, ISHELL AND RSHELL, NEITHER REQUIRING ASCEND, ISHELL SORTS
C AN ARRAY OF TYPE INTEGER, RSHELL SORTS AN ARRAY OF TYPE REAL. (SHELL
C CAN DO THE SAME JOB, BUT IS SLOWER.)
C
C FOR PURPOSES OF DESCRIPTION, LET US CALL THE KEYS X(1),X(2),...,X(N).
C RATHER THAN MOVE THE X'S AROUND, THE ROUTINE USES THE ARRAY IRANK TO
C INDICATE THE ORDER, AS FOLLOWS... UPON RETURN TO THE CALLING PROGRAM,
C IRANK(1) CONTAINS THE SUBSCRIPT OF THE FIRST X, IRANK(2) CONTAINS THE
C SUBSCRIPT OF THE SECOND X, ..., IRANK(N) CONTAINS THE SUBSCRIPT OF THE
C LAST X. THE X ARRAY IS NOT ALTERED IN ANY WAY.
C
C EXAMPLE...
C INPUT      N = 4
C            X = 519, 857, 103, 776
C OUTPUT     IRANK = 3, 1, 4, 2
C
C A PRECISE DESCRIPTION OF THE RESULTS IS...
C IF J.LT.K, THEN X(JJ).LE.X(KK), WHERE JJ=IRANK(J) AND KK=IRANK(K).
C
C THE X'S MAY BE POSITIVE OR NEGATIVE, THERE IS NO RESTRICTION ON N,
C EXCEPT THAT IF N.LT.1 NO ACTION TAKES PLACE*
C
C SHELL'S METHOD IS DESCRIBED IN THE COMMUNICATIONS OF THE A. C. M.,
C MAY 1963, IN ARTICLES BY GOTLIER AND BY HIBBARD, THE PRESENT ROUTINE
C IS BASED ON HIBBARD'S ALGOL ALGORITHM ON PAGE 209. ANOTHER ALGOL
C ALGORITHM, BY J. BOOTHROYD, CAN BE FOUND IN COMMUNICATIONS OF THE
C A. C. M., AUGUST 1963, PAGE 445 (ALGORITHM 201).
C
C THE TIME REQUIRED FOR SORTING UNIFORMLY RANDOM KEYS IS ROUGHLY
C PROPORTIONAL TO N*LOG2(N), WHERE LOG2 IS THE LOGARITHM BASE 2,
C TIME = .16*N*LOG2(N) MILLISECONDS (IBM 7094 MODEL 1)
C
C THIS PROGRAM WAS WRITTEN IN ASA STANDARD FORTRAN BY ROBERT M. RUSSELL,
C MAY 1967,
C LOGICAL ASCEND
C INTEGER IRANK(N)
C IF(N.LE.0) GO TO 100
C DO 10 J=1,N
C 10 IRANK(J)=J
C IF(N.EQ.1) GO TO 100

```

```

SHELL
JR T
JRT
JRT
JRT
JRT
JRT
SHELL 3
SHELL 4
SHELL 5
SHELL 6
SHELL 7
SHELL 8
SHELL 9
SHELL 10
SHELL 11
SHELL 12
SHELL 13
SHELL 14
SHELL 15
SHELL 16
SHELL 17
SHELL 18
SHELL 19
SHELL 20
SHELL 21
SHELL 22
SHELL 23
SHELL 24
SHELL 25
SHELL 26
SHELL 27
SHELL 26
SHELL 29
SHELL 30
SHELL 31
SHELL 32
SHELL 33
SHELL 34
SHELL 35
SHELL 36
SHELL 37
SHELL 3F
SHELL 39
SHELL 40
SHELL 41
SHELL 42
SHELL 43
SHELL 44
SHELL 45
SHELL 46
SHELL 47
SHELL 48
SHELL 49
SHELL 50
SHELL 51
SHELL 52

```

```

      I=1
20  M=M+M
    IF (M,LT,N) GO TO 20
    M=MAX0(1,M/2-1)
30  I=1
    IM=I+M
40  J=I
    JM=IM
    IHOLD=IRANK(IM)
50  JR=IRANK(J)
    IF (ASCENS(X,JR,IHOLD)) GO TO 60
--- IF (ASCEND(X,JR,IHOLD)) GO TO 60
    IRANK(JM)=IRANK(J)
    JM=J
    J=J-M
    IF (J,GT,0) GO TO 50
60  IRANK(JM)=IHOLD
    I=I+1
    IM=I+M
    IF (IM,LE,N) GO TO 40
    M=M/2
    IF (M,GT,0) GO TO 30
100 RETURN
    END

```

```

SHELL 53
SHELL 54
SHtLL 55
SHtLL 56
SHELL 57
SHELL 58
ShELL 59
SHELL 60
SHELL 61
SHELL 62
SHtLL 63
SHELL 63
SHtLL 64
SHELL 65
SHELL 66
SHELL 67
ShELL 68
SHELL 69
SHELL 70
SHELL 71
SHELL 72
SHELL 73
SHELL 74
SHELL 75

```


C-----	SUBROUTINE SHELLV(X,IRANK,N)	SHELL 2
C-----		JRT
C-----	SUBROUTINE SHELL CHANGED TO SHELLV BY JOHN R. TADLOCK NOV. 1975.	JHT
C-----		JRT
C-----	FOR AN EXPLANATION OF HOW THIS SUBROUTINE WORKS REFER TO	JRT
C-----	COMMENTS CON SUBROUTINE SHELLS	
C----	THERE ARE 3 VERSIONS OF SWELL AND ASCEND IN THE FAMULUS LIBRARY	JRT
C----	INDEX CALLS SUBROUTINE SHELLI WHICH CALLS FUNCTION ASCEND	JRT
C----	SORT CALLS SUBROUTINE SHELLS WHICH CALLS FUNCTION ASCEND	JRT
C----	VOCAP CALLS SUBROUTINE SHELLV WHICH CALLS FUNCTION ASCENV	JRT
C----		JRT
C		SHELL 44
C	THIS PROGRAM WAS WRITTEN IN ASA STANDARD FORTRAN BY ROBERT M. RUSSELL,	SHELL 45
C	MAY 1967.	SHELL 46
C		SHELL 40
	LOGICAL ASCENV	SHELL 47
	INTEGER IRANK(N)	SHELL 48
	IF(N.LE.0) GO TO 100	SHELL 49
	DO 10 J=1,N	SHELL 50
10	IRANK(J)=J	SHELL 51
	IF(N.EQ.1) GO TO 100	SHELL 52
	M=1	SHELL 53
20	M=M+M	SHELL 54
	IF(M.LT.N) GO TO 20	SHELL 55
	M=MAX(1,M/2-1)	SHELL 56
30	I=1	SHELL 57
	IM=I+M	SHELL 58
40	J=I	SHELL 59
	JM=IM	SHELL 60
	IHOLD=IRANK(IM)	SHELL 61
50	JR=IRANK(J)	SHELL 62
	IF(ASCENV(X,JR,IHOLD)) GO TO 60	SHELL 63
	IRANK(JM)=IRANK(J)	SHELL 64
	JM=J	SHELL 65
	J=J-M	SHELL 66
	IF(J.GT.0) GO TO 50	SHELL 67
60	IRANK(JM)=IHOLD	SHELL 68
	I=I+1	SHELL 69
	IF=I+M	SHELL 70
	IF(IM.LE.0) GO TO 40	SHELL 71
	M=M/2	SHELL 72
	IF(M.GT.0) GO TO 30	SHELL 73
100	RETURN	SHELL 74
	END	SHELL 75

SUBROUTINE SIMILE(KON,X,LX,NX,Y,LY,NY,LIKE)	SIMILE 2
C GIVEN TWO CHARACTER SUBSTRINGS, SIMILE COMPARES THEM TO SEE IF THEY	SIMILE 3
C MATCH.	SIMILE 4
C THE TWO SUBSTRINGS ARE X AND Y, X CONSISTS OF NX CHARACTERS BEGINNING	SIMILE 5
C IN POSITION LX +1 OF X, AND Y CONSISTS OF NY CHARACTERS BEGINNING	SIMILE 6
C IN POSITION LY+1 OF Y.	SIMILE 7
C FOR A MATCH TO BE EFFECTED, WHEN KON=0, IT IS NECESSARY THAT THE FIRST	SIMILE 8
C NX CHARACTERS OF THE Y SUBSTRING MATCH THE ENTIRE X SUBSTRING.	SIMILE 9
C WHEN KON=1, AN ADDITIONAL REQUIREMENT IS THAT NX=NY.	SIMILE10
C IF NX IS GREATER THAY NY, NO MATCH IS EVER POSSIBLE,	SIMILE11
C THE RESULT IS INDICATED BY LIKE. LIKE=1 SAYS THAT THE SUBSTRINGS	SIMILE12
C MATCH, LIKE=0 SAYS THEY DO NOT,	SIMILE13
C IF NX=0, OR NY=0, OR NX IS GREATER THAN NY, THEN LIKE=0.	SIMILE14
C FOR SPEED, IT IS ADVISABLE TO HAND CODE THIS SUBROUTINE.	SIMILE15
INTEGER X(2),Y(2)	SIMILE16
LIKE=0	SIMILE17
IF(NX.GT.NY .OR. NX.LE.0) RETURN	SIMILE18
IF(KON.EQ.1 .AND. NX.NE.NY) RETURN	SIMILE19
JX=LX	SIMILE20
JY=LY	SIMILE21
DO 10 J=1,NX	SIMILE22
JX=JX+1	SIMILE23
JY=JY+1	SIMILE24
IF(KHAR(X,JX).NE.KHAR(Y,JY)) RETURN	SIMILE25
10 CONTINUE	SIMILE26
LIKE=1	SIMILE27
RETURN	SIMILE28
END	SIMILE29

C	SUBROUTINE SYMBOL	SYMBOL 2
C	THIS SUBROUTINE COLLECTS SYMBOLS OF THE FAMULUS CONTROL LANGUAGE FROM	SYMBOL 3
C	THE INPUT CARDS, EACH CALL SYMBOL STATEMENT WHEN EXECUTED COLLECTS	SYMBOL 4
C	THE NEXT SYMBOL AND STORES INDICATIONS OF IT IN /INPT/ COMMON,	SYMBOL 5
C		SYMBOL 6
C	LIT SYMBOL	SYMBOL 7
C	1 ALPHAMERIC (DESCRIPTOR, TERM, FIELD LABEL)	SYMBOL 8
C	2 INTEGER	SYMBOL 9
C	3 /	SYMBOL 10
C	4 (	SYMBOL 11
C	5	SYMBOL 12
C	6 ,	SYMBOL 13
C	7 -	SYMBOL 14
C	8 *	SYMBOL 15
C	9 :	SYMBOL 16
C	10 %	SYMBOL 17
C	11 %	SYMBOL 18
C	12 END OF STATEMENT	SYMBOL 19
	COMMON /CORE/NCHW,BLANKS,BLANK	SYMBOL 20
	INTEGER BLANKS,BLANK	SYMBOL 21
	COMMON /DSCR/IDSCR	SYMBOL 22
	COMMON /FAM/SLASH,STAR,COMMA,ZERO,MXCHAR,MXCODES,MXWDES,MXCHW,	SYMBOL 23
	MXFREE,MXWFRE	SYMBOL 24
	INTEGER SLASH,STAR,COMMA,ZERO	SYMBOL 25
	COMMON /INPT/CARD(10),KOLUM,HOLD,ENDCAR,BLACAR,CONCAR,ERRIN,NWC,	SYMBOL 26
	MXFL,MXCAR,NCARIN,KHR,KIN,LIT,LITX,SYM(5),NUMB,NCSYM,CATLOG,	SYMBOL 27
	KARA,KARB	SYMBOL 28
	INTEGER CARD,SYM	SYMBOL 29
	LOGICAL HOLD,ENDCAR,BLACAR,CONCAR,ERRIN,CATLOG	SYMBOL 30
	INTEGER MAP(13),CH(12)	SYMBOL 31
C	MAP(13) RELATES ONE-CHAR SYMBOLS WITH THEIR VALUE FROM IDEAL	SYMBOL 32
C	FUNCTION, IF IDEAL (KHR)=KIN, THEN MAP(KIN)=LIT, THE	SYMBOL 33
C	TYPE OF SYMBOL IT IS,	SYMBOL 34
	DATA (MAP(J),J=1,13)/3*0,3,4,5,6,7,8,9,10,11,0/	SYMBOL 35
	DATA(CH(J),J=1,12)/1H ,1H ,1H/,1H(,1H),1H,,1H-,1H*,1H!,1H&,1H%,	SYMBOL 36
	1H /	SYMBOL 37
C	CALL SPOT(LIT,LITX,KHR,KIN,4HIIAI,24HLIT LITX KHR KIN)	SYMBOL 38
C	CALL SPOT(NUMB,NCSYM,NCARIN,KOLUM,4HIIII,	SYMBOL 39
C	24HNUMB NCSYM NCARINKOLUM)	SYMBOL 40
C	CALL SPOT(ERRIN,ENDCAR,CONCAR,HOLD,4HLLLL,	SYMBOL 41
C	24HERRIN ENDCARCONCARHOLD)	SYMBOL 42
	LITX=LIT	SYMBOL 43
	NUMB=0	SYMBOL 44
	NCSYM=0	SYMBOL 45
	CALL MOVE(BLANKS,SYM,-MXWDES)	SYMBOL 46
C	FIND FIRST NON-BLANK COLUMN,	SYMBOL 47
	10 IF(ENDCAR) GO TO 70	SYMBOL 48
	IF(KIN,NE,1) GO TO 15	SYMBOL 49
	CALL GETCH	SYMBOL 50
	GO TO 10	SYMBOL 51
	15 IF(IDSCR,EQ,1,AND,KIN,EQ,3) GO TO 20	SYMBOL 52
	IF(KIN,EQ,2) GO TO 20	SYMBOL 53
	IF(KIN,EQ,3) GO TO 40	SYMBOL 54
	GO TO 60	SYMBOL 55
C	INITIALIZE ALPHAMERIC SYMBOL,	SYMBOL 56
C	IF /DSCR/IDSCR=1, AN ALPHAMERIC SYMBOL MAY BEGIN WITH A DIGIT AND	SYMBOL 57
C	KAY INCLUDE ANY CHARACTER K SUCH THAT IDEAL(K)=13,	SYMBOL 58

20 LIT=1	SYMBOL59
N=0	SYMBOL60
C PRUCSS CURRENT CHARACTER IN ALPHAMERIC SYMBOL.	SYMBOL61
30 KE=1	SYMBOL62
IF(N,GE,MXCDER,AND,KIN,NE,1) GO TO 900	SYMBOL63
IF(N,GT,0,OR,KIN,NE,1) N=N+1	SYMBOL64
IF(N,LE,MXCDER) CALL CHAR(KHR,SYM,N)	SYMBOL65
IF(KIN,NE,1) NCSYM=N	SYMBOL66
C GET NEXT CHARACTER IN ALPHAMERIC SYMBOL.	SYMBOL67
LALL GETCH	SYMBOL68
IF(ENDCAR) GO TO 100	SYMBOL69
IF(KIN,GE,1,AND,KIN,LE,3,OR,KIN,EQ,8) GO TO 30	SYMBOL70
IF(IDSCRIF,EQ,1,AND,KIN,EQ,13) GO TO 30	SYMBOL71
GO TO 100	SYMBOL72
C INITIALIZE INTEGER SYMBOL.	SYMBOL73
40 LIT=2	SYMBOL74
N=0	SYMBOL75
NUMB=0	SYMBOL76
C PROCESS CURRENT CHARACTER IN INTEGER SYMBOL.	SYMBOL77
50 N=N+1	SYMBOL78
KE=2	SYMBOL79
IF(N,GT,10) GO TO 900	SYMBOL80
IF(N,LE,10) NUMB=10*NUMB+KHR-ZERO	SYMBOL81
CALL CHAR(KHR,SYM,N)	SYMBOL82
NCSYM=N	SYMBOL83
C GET NEXT CHARACTER IN INTEGER SYMBOL.	SYMBOL84
CALL GETCH	SYMBOL85
IF(KIN,EQ,3) GO TO 50	SYMBOL86
GO TO 100	SYMBOL87
C SYMBOL CONSISTING OF A SINGLE CHARACTER.	SYMBOL88
60 LIT=MAP(KIN)	SYMBOL89
NCSYM=1	SYMBOL90
KE=3	SYMBOL91
IF(LIT,EQ,0) GO TO 900	SYMBOL92
SYM(1)=CH(LIT)	SYMBOL93
IF(LIT,EQ,3,AND,KOLUM,EQ,1,AND,LITX,NE,12)GO TO 70	SYMBOL94
CALL GETCH	SYMBOL95
GO TO 100	SYMBOL96
C END-OF-STATEMENT SYMBOL.	SYMBOL97
70 LIT=12	SYMBOL98
GO TO 110	SYMBOL99
C MAKE SURE NEXT CHARACTER IS NON-BLANK.	SYMBOL100
100 IF(ENDCAR) GO TO 110	SYMBOL101
IF(KIN,NE,1) GO TO 110	SYMBOL102
CALL GETCH	SYMBOL103
GO TO 100	SYMBOL104
C FINISH	SYMBOL105
110 CONTINUE	SYMBOL106
CALL TALMUD(6HSYMBOL,SYM,0,MXCDER)	SYMBOL107
C CALL SPOT(LIT,LITX,KHR,KIN,4HIIAI,24HLIT LITX KHR K14)	SYMBOL108
C CALL SPOT(NUMB,NCSYM,NCARIN,KOLUM,4HIIII,	SYMBOL109
C , 24HNUMB NCSYY NCARINKOLUM)	SYMBOL110
C CALL SPOT(ERRIN,ENDCAR,CONCAR,HOLD,4HLLLL,	SYMBOL111
C , 24HERRIN ENDCARCONCARHOLD)	SYMBOL112
RETURN	SYMBOL113
C ERROR MESSAGE.	SYMBOL114
900 CALL DIVEST	SYMBOL115

```

      CALL PAGER(2)
      PRINT 1000,KOLUM
1000  FORMAT(30H ERROR IN ABOVE CARD AT COLUMN,13,10H OH BEYOND,3X,
      ,      82(1H*))
      PRINT 9050, KE
9050  FORMAT(118X,6HSYMBOL 14)
      ERRIN=.TRUE,
      RETURN
      END

```

```

SYMB0116
SYMB0117
SYMB0118
SYMB0119
SYMB0120
SYMB0121
SYMB0122
SYMB0123
SYMB0124

```

SUBROUTINE TALMUD(TAG,X,LX,NX)	TALMUD 2
C THIS IS A DEBUGGING SUBROUTINE, USED TO PRINT THE CONTENTS OF A	TALMUD 3
C CHARACTER STRING,	TALMUD 4
C THE CHARACTER STRING IS X, AND POSITIONS LX+1 THRU LX+NX (INCLUSIVE)	TALMUD 5
C ARE PRINTED.	TALMUD 6
C TAG IS AN 8-CHARACTER LABEL TO PRINT WITH IT.	TALMUD 7
COMMON /CORE/NCHW,BLANKS,BLANK	TALMUD 8
INTEGER BLANKS,BLANK	TALMUD 9
COMMON /TEST/ITEST	TALMUD10
INTEGER TAG(1),X(2),LINE(13)	TALMUD11
IF(ITEST.EQ.0) RETURN	TALMUD12
IF(NX.GE.1) GO TO 10	TALMUD13
C NO TEXT TO PRINT.	TALMUD14
CALL PAGER(1)	TALMUD15
PRINT 1000, TAG,LX,NX	TALMUD16
1000 FORMAT(1X,A8,2I5,2X,13A8)	TALMUD17
RETURN	TALMUD18
C PRINT TEXT,	TALMUD19
10 L=LX	TALMUD20
N=MIN0(NX,10000)	TALMUD21
LAST=LX+N	TALMUD22
20 N=MIN0(LAST-L,100)	TALMUD23
DO 30 J=1,13	TALMUD24
30 LINE(J)=BLANKS	TALMUD25
CALL CHOP(X,L,LINE,0,M)	TALMUD26
CALL PAGER(1)	TALMUD27
IF(L.EQ.LX) PRINT 1000,TAG,LX,NX,LINE	TALMUD28
IF(L.NE.LX) PRINT 2000,LINE	TALMUD29
2000 FORMAT(21X,13A8)	TALMUD30
L=L+M	TALMUD31
IF(L.LT.LAST) GO TO 20	TALMUD32
RETURN	TALMUD33
END	TALMUD34

SUBROUTINE TAPER(MT)	TAPER 2
C USED FOR DEBUGGING, TAPER HEADS CARD CITATIONS IN SPECIAL FORMAT VIA	TAPER 3
C INGEST TESTING SUBROUTINE AND WRITES A TAPE (UNIT MT).	TAPER 4
C ALSO HEADS THE INFORMATION NEEDED FOR A PREFACE RECORD.	TAPER 5
C THE CARDS TAPER HEADS ARE LOCATED FOLLOWING THE /DEBUG/ STATEMENT AND	TAPER 6
C BEFORE THE /ID/ STATEMENT,	TAPIR 7
C THE PROCEDURE IS AS FOLLOWS...	TAPER 8
C THE FIRST CARD IS ID.	TAPEH 9
C THE SECOND CARD CONTAINS FIELD LABELS, IN A5 FORMAT,	TAPIR 10
C THEN A NUMBER OF CARD CITATIONS ARE READ VIA INGEST.	TAPER 11
C FINALLY A CARD WITH ONLY \$ IN COLUMN 1 TERMINATES THIS SPECIAL INPUT,	TAPER 12
C AT THIS POINT TAPE MT IS READY TO BE USED AS INPUT TO THE PROGRAM	TAPIR 13
C BEING TESTED,	TAPER 14
COMMON /CORE/NCHW,BLANKS,BLANK	TAPER 15
INTEGER BLANKS,BLANK	TAPER 16
COMMON /FAM/SLASH,STAR,COMMA,ZERO,MXCHAR,MXCDER,MXWDES,MXCHW,	TAPER 17
MXFREE,MXWFE	TAPER 18
INTEGER SLASH,STAR,COMMA,ZERO	TAPER 19
COMMON /FIEL/FIELD(10),IFIELD(10),NFIELD,KDESCR	TAPER 20
INTEGER FIELD	TAPER 21
COMMON /INPT/CARD(10),KOLUM,HOLD,ENDCAR,BLACAR,CONCAR,ERRIN,NWC,	TAPEH 22
MXFL,MAXCAR,NCARIN,KHR,KIN,LIT,LITX,SYM(5),NUMB,NCSYM,CATLOG,	TAPER 23
KARA,KARB	TAPER 24
INTEGER CARD,SYM	TAPER 25
LOGICAL HOLD,ENDCAR,BLACAR,CONCAR,ERRIN,CATLOG	TAPER 26
COMMON /TEST/ITEST	TAPER 27
INTEGER ID(13),DESC,CIT(450)	TAPER 28
INTEGER FLD(10),IFLD(10)	TAPER 29
LOGICAL MATCH	TAPER 30
DATA DESC/4HDESC/	TAPER 31
CALL BUTLER	TAPER 32
ITST=ITEST	TAPER 33
IF(ITST.NE.2) ITEST=0	TAPER 34
IF(CONCAR) GO TO 100	TAPIR 35
C SAVE /FIEL/.	TAPER 36
NFLD=NFIELD	TAPER 37
KDES=KDESCR	TAPER 38
GO 5 K=1,10	TAPEH 39
FLD(K)=FIELD(K)	TAPER 40
5 IFLD(K)=IFIELD(K)	TAPER 41
C READ ID AND FIELD INFORMATION.	TAPER 42
CALL PAGER(-3)	TAPER 43
CALL DIVEST	TAPER 44
CALL MOVE(BLANKS,ID,-MXWFE)	TAPER 45
CALL MOVE(CARD,ID,NWC)	TAPER 46
CALL TALMUD(2HID,ID,0,MXFREE)	TAPER 47
CALL INCARD	TAPER 48
DO 7 K=1,10	TAPEH 49
FIELD(K)=BLANKS	TAPER 50
7 CALL CHOP(CARD,5*(K-1),FIELD(K),0,MXFL)	TAPER 51
KDESCR=0	TAPER 52
DO 10 K=1,10	TAPER 53
IF(MATCH(FIELD(K),BLANKS)) GO TO 20	TAPIR 54
CALL TALMUD(5HFIELD,FIELD(K),0,MXFL)	TAPER 55
IFIELD(K)=K	TAPER 56
IF(MATCH(FIELD(K),DESC)) KDESCR=K	TAPER 57
10 CONTINUE	TAPER 58


```

      K=11
20  NFIELD=K-1
      CALL REWAND(MT)
C WRITE PREFACE YECOH,
      CALL IOIL(1,MT,ID,FIELD,IFIELD,NFIELD,KDESCR)
      CALL IDPRIN(ID,FIELD,IFIELD,NFIELD,KDESCR,
, 20,20HDEBUGGING INPUT TAPE)
C READ CITATIONS.
      L=0
40  CALL INGEST(CIT, LAST)
      L=L+1
      IF(LAST.EQ.0) CALL PRNCIT(CIT,L,0,128)
      CALL IOTAPE(1,MT,CIT, LAST)
      IF(LAST.EQ.0) GO TO 40
      CALL REWAND(MT)
      CALL INCARD
C RESET /FIEL/.
      NFIELD=NFLD
      KDESCR=KDES
      DO 50 K=1,10
      FIELD(K)=FLD(K)
50  IFIELD(K)=IFLD(K)
      IF(CONCAR) CALL PAGER(-3)
100 CONTINUE
      ITST=ITST
      RETURN
      END

```

```

TAPER 59
TAPER 60
TAPER 61
TAPER 62
TAPER 63
TAPER 64
TAPER 65
TAPER 66
TAPER 67
TAPER 68
TAPER 69
TAPER 70
TAPER 71
TAPER 72
TAPER 75
TAPER 74
TAPER 75
TAPER 76
TAPER 77
TAPER 78
TAPER 79
TAPER 80
TAPER 81
TAPER 82
TAPER 83
TAPER A4
TAPER 85

```

SUBROUTINE TINKER	TINKER 2
C READS INPUT FROM /REPLACE/ STATEMENT AND STORES IT IN CONVENIENT FORM,	TINKER 3
C THUS, IT SETS NREP AND THE ARRAYS LREP,KREP,PATREP,LOCPAT,NCHPAT,	TINKER 4
C LOCREP, AND NCHREP.	TINKER 5
COMMON /EDT/NREP,LREP(2,100),KREP(100),PATREP(800),LOCPAT(100),	TINKER 6
NCHPAT(100),LOCREP(100),NCHREP(100)	TINKER 7
INTEGER PATREP	TINKER 8
COMMON /CORE/NCHW,BLANKS,BLANK	TINKER 9
INTEGER BLANKS,BLANK	TINKER10
COMMON /FAM/SLASH,STAR,COMMA,ZERO,MXCHAR,MXCODES,MXWDES,MXCHW,	TINKER11
MXFREE,MXWFRE	TINKER12
INTEGER SLASH,STAR,COMMA,ZERO	TINKER13
COMMON /FIEL/FIELD(10),IFIELD(10),NFIELD,KDESCR	TINKER14
INTEGER FIELD	TINKER15
COMMON /INPT/CARD(10),KOLUM,HOLD,ENDCAR,BLACAR,CONCAR,ERRIN,NWC,	TINKER16
MXFL,MAXCAR,NCARIN,KHR,KIN,LIT,LITX,SYM(5),NUMB,NCSYM,CATLOG,	TINKER17
KARA,KARB	TINKER18
INTEGER CARD,SYM	TINKER19
LOGICAL HOLD,ENDCAR,BLACAR,CONCAR,ERRIN,CATLOG	TINKER20
INTEGER NUM(3)	TINKER21
C INITIALIZE.	TINKER22
CALL BUTLER	TINKER23
JYR=0	TINKER24
LPR=0	TINKER25
NCAR=NCARIN	TINKER26
C MAXIMUM NUMBER OF CHARACTERS IN ALL PATTERN-REPLACEMENT TEXT (I.E., IN	TINKER27
C PATREP) IS MXREP.	TINKER28
MXREP=800*NCHW	TINKER29
C PROCESS INPUT CARDS FOR NEXT PATTERN-REPLACEMENT PAIR.	TINKER30
10 IF(JPR,GE,100) GO TO 902	TINKER31
JPR=JPR+1	TINKER32
CALL GETNUM(NUM,1,2,N,KERR)	TINKER33
IF(KERR,GT,0) GO TO 901	TINKER34
LREP(1,JPR)=NUM(1)	TINKER35
LREP(2,JPR)=IABS(NUM(N))	TINKER36
CALL GETLAB(LAB,1,1,N,KERR)	TINKER37
IF(KERR,GT,0) GO TO 901	TINKER38
KREP(JPR)=LAB	TINKER39
CALL SPOT(JPR,LREP(1,JPR),LREP(2,JPR),KREP(JPR),4HIIIA,	TINKER40
24HJPR LREP 1LREP 2KREP)	TINKER41
C COLLECT PATTERN AND REPLACEMENT CHARACTERS.	TINKER42
IF(KHR,NE,STAR) GO TO 906	TINKER43
DO 40 JAMES=1,2	TINKER44
IF(JAMES,EQ,1) LOCPAT(JPR)=LPR	TINKER45
IF(JAMES,EQ,2) LOCREP(JPR)=LPR	TINKER46
LP=LPR	TINKER47
20 CALL GETCH	TINKER48
IF(KHR,EQ,STAR) GO TO 30	TINKER49
IF(ENDCAR) GO TO 904	TINKER50
IF(NCARIN,GT,NCAR,AND,CONCAR) GO TO 906	TINKER51
IF(LPR,GE,MXREP) GO TO 905	TINKER52
LPR=LPR+1	TINKER53
CALL CHAR(KHR,PATREP,LPR)	TINKER54
GO TO 20	TINKER55
30 NPR=LPR-LP	TINKER56
CALL TALMUD(6HPATREP,PATREP,LP,NPR)	TINKER57
IF(JAMES,EQ,1) NCHPAT(JPR)=NPR	TINKER58

IF(JAMES,F11-2) NCHREP(JPR)=NPR	TINKER59
40 CONTINUE	TINKER60
C LAST CHARACTER READ WAS THE THIRL ASTERISK.	TINKER61
C RESET SYMBOL COLLECTOR.	TINKER62
LIT=8	TINKER63
CALL CHOP(BLANKS,0,SYM,0,-MXCDES)	TINKER64
CALL CHAR(STAR,SYM,1)	TINKER65
50 CALL GETLH	TINKER66
IF(ENDCAR) GO TO 100	TINKER67
IF(KIN.EG,1) GO TO 50	TINKER68
IF(KIN.EG,5) GO TO 10	TINKER69
IF(KIN.NE,4) GO TO 901	TINKER72
C NORMAL RETURN.	TINKER71
100 NREP=JPR	TINKER72
CALL KORAN(6HLOCPAT,LOCPAT,NREP)	TINKER73
CALL KORAN(6HNCHPAT,NCHPAT,NREP)	TINKER74
CALL KORAN(6HLOCREP,LOCREP,NREP)	TINKER75
CALL KORAN(6HNCHREP,NCHREP,NREP)	TINKER76
RETURN	TINKER77
C ERRORS.	TINKER78
901 KE=1	TINKER79
GO TO 910	TINKER80
902 KE=2	TINKER81
GO TO 910	TINKER82
904 KE=4	TINKER83
GO TO 910	TINKER84
905 KE=5	TINKER85
GO TO 910	TINKER86
906 KE=6	TINKER87
910 CALL SCOUT(0,0,BLANK,CARD,0,KOLUM,N)	TINKER88
IF(N.LT,KOLUM-1) CALL DIVEST	TINKER89
CALL PAGER(1)	TINKER90
GO TO (921,922,930,924,925,926),KE	TINKER91
C SYNTAX ERROR.	TINKER92
921 PRINT 9001, SYM	TINKER93
9001 FORMAT(34H SYNTAX ERROR INVOLVING THE SYMBOL, 5A8,5X,49(1H*))	TINKER94
GO TO 930	TINKER95
C TOO MANY REPLACEMENTS.	TINKER96
922 PRINT 9002, JPR	TINKER97
9002 FORMAT(54H YOU PAVE EXCEEDED THE MAXIMUM NUMBER OF REPLACEMENTS,,	TINKER98
14,5X,65(1H*))	TINKER99
GO TO 930	TINKE100
C SUDDEN END OF INPUT.	TINKE101
924 PRINT 9004	TINKE102
9004 FORMAT(20H ABRUPT END OF INPUT, 5X,103(1H*))	TINKE103
GO TO 930	TINKE104
C TOO MUCH PATTERN-REPLACEMENT TEXT.	TINKE105
925 PRINT 9005, LPH	TINKE106
9005 FORMAT(87H YOU HAVE EXCEEDED THE MAXIMUM NUMBER OF CHARACTERS IN A	TINKE107
LL THE REPLACEMENT TEXT, NAMELY, 15,5X,31(1H*))	TINKE108
CALL PAGES(1)	TINKE109
PRINT 9055	TINKE110
9055 FORMAT(33H I SUPPOSE YOU FORGOT AN ASTERISK, 5X,90(1H*))	TINKE111
GO TO 930	TINKE112
926 PRINT 9006	TINKE113
9006 FORMAT(29H ERROR IN /REPLACE/ STATEMENT,5X,94(1H*))	TINKE114
C END OF ERROR ANALYSIS.	TINKE115
930 CALL PAGER(-1)	TINKE116
ERRIN=.TRUE.	TINKE117
RETURN	TINKE118
END	TINKE119

SUBROUTINE USHER(IDEN,KERR)	USHER 2
C PERFORMS CERTAIN TASKS NECESSARY AT THE BEGINNING OF MOST FAMULUS	USHER 3
C MAIN PROGRAMS. AMONG THEM ARE, INITIALIZING PAGER, SETTING FAMULUS	USHER 4
C COMMON VALUES, READING AND PROCESSING THE /ID/ STATEMENT, ALSO	USHER 5
C ACCEPTS A /DEBUG/ STATEMENT BEFORE /ID/.	USHER 6
C A NUMBER LIST OF LENGTH 1 ON THE /DEBUG/ CAHD SETS ITEST TO THAT	USHER 7
C VALUE. IF NO NUMBER LIST, SETS ITEST=1,	USHER 8
C ERROR CONDITIONS ARE INDICATED BY RETURNING A NONZERO VALUE OF KEWR,	USHER 9
C ZERO IS RETURNED IF NO ERHOR,	USHER 10
C THE TEXT OF /ID/ IS PLACED IN THE ARRAY IDEN AND ALSO IN PAGHED.	USHER 11
COMMON /FAM/SLASH,STAR,COMMA,ZERO,MXCHAR,MXCDER,MXWDER,MXCHW,	USHER 12
MXFREE,MXWFRE	USHER 13
INTEGER SLASH,STAR,COMMA,ZERO	USHER 14
COMMON /INPT/CARD(10),KOLUM,HOLD,ENDCAR,BLACAR,CONCAR,ERRIN,NWC,	USHER 15
MXFL,MXCAR,NCARIN,KHR,KIN,LIT,LITX,SYM(5),NUMB,NCSYM,CATLOG,	USHER 16
KARA,KARB	USHER 17
INTEGER CARD,SYM	USHER 18
LOGICAL HOLD,ENDCAR,BLACAR,CONCAR,ERRIN,CATLOG	USHER 19
COMMON /PAGING/IPAGE,LINES,MAXLIN,PAGHED(13)	USHER 20
COMMON /TEST/ITEST	USHER 21
INTEGER IDEN(2),LIST(2)	USHER 22
ITEST=0	USHER 23
CALL SFTCOM	USHER 24
CALL PAGER(0)	USHER 25
CALL SETSYM	USHER 26
IF(ENDCAR) GO TO 90	USHER 27
	USHER 28
CALL FABRIC(KON)	USHER 29
	USHER 30
IF(KON.NE.12) GO TO 10	USHER 31
LIST(1)=1	USHER 32
IF(CONCAR.AND.KOLUM.NE.1) CALL GETNUM(LIST,1,1,N,KER)	USHER 33
ITEST=LIST(1)	USHER 34
CALL TAPFR(1)	USHER 35
CALL TAPER(2)	USHER 36
CALL SETSYM	USHER 37
IF(ENDCAR) GO TO 90	USHER 38
CALL FABRIC(KON)	USHER 39
10 IF(ERRIN.OR.KON.NE.1) GO TO 90	USHER 40
CALL GOSSIP(IDEN)	USHER 41
IF(ERRIN) GO TO 93	USHER 42
CALL CHOP(IDEN,0,PAGHED,0,MXFREE)	USHER 43
KERR=0	USHER 44
GO TO 100	USHER 45
90 KERR=1	USHER 46
100 RETURN	USHER 47
END	USHER 48

SUBROUTINE VSTAT(NT,T)	VSTAT 2
C PRINTS STATISTICS ON THE DICTIONARY IN VOCAB.	VSTAT 3
C T IS A TITLE TO PRINT. NT IS THE NUMBER OF CHARACTERS IN T.	VSTAT 4
COMMON /FAM/SLASH,STAR,COMMA,ZERO,MXCHAR,MXCDES,MXWDES,MXCHW,	VSTAT 5
, MXFREE,MXWFRE	VSTAT 6
INTEGER SLASH,STAR,COMMA,ZERO	VSTAT 7
COMMON NDICT,NVOC,NTRIV,MXHASH,MXENT,MXDICT,WORD(5),	VSTAT 8
, DICT(6750),LDICT(6501),HASH(9511)	VSTAT 9
INTEGER DICT,HASH,WORD,KRANK(6500)	VSTAT 10
EQUIVALENCE(KRANK,WASH)	VSTAT 11
INTEGER T(2)	VSTAT 12
INTEGER LENTH(40)	VSTAT 13
C LENTH(J) IS THE NUMBER OF ENTRIES WHICH HAVE J CHARACTERS. THE	VSTAT 14
C DIMENSION OF LENTH IS MXCDES.	VSTAT 15
NNT=(NT+MXCHW-1)/MXCHW	VSTAT 16
CALL PAGER(4)	VSTAT 17
PRINT 1000,(T(J),J=1,NNT)	VSTAT 18
1000 FORMAT(/16H STATISTICS FOR ,10A10/1X)	VSTAT 19
IF(NTRIV.LE.0) GO TO 30	VSTAT 20
C TRIVIAL WORDS	VSTAT 21
CALL PAGER(1)	VSTAT 22
PRINT 2000	VSTAT 23
2000 FORMAT(17H TRIVIAL WORDS...)	VSTAT 24
CALL MOVE(0,LENTH,-MXCDES)	VSTAT 25
DO 10 K=1,NDICT	VSTAT 26
IF(ISIGN(1,LDICT(K)).GT.0) GO TO 13	VSTAT 27
D=IABS(LDICT(K+1))-IABS(LDICT(K))	VSTAT 28
LENTH(ND)=LENTH(ND)+1	VSTAT 29
10 CONTINUE	VSTAT 30
MOST=0	VSTAT 31
MEAN=0	VSTAT 32
DO 20 J=1,MXCDES	VSTAT 33
IF(LENTH(J).NE.0)MOST=J	VSTAT 34
20 MEAN=MEAN+J*LENTH(J)	VSTAT 35
CALL PAGER(6)	VSTAT 36
PRINT 2001,(LENTH(J),J=1,MOST)	VSTAT 37
2001 FORMAT(29H DISTRIBUTION OF WORD LENGTHS/(1015))	VSTAT 38
PRINT 2002,NTRIV	VSTAT 39
2002 FORMAT(26H NUMBER OF TRIVIAL WORDS =15)	VSTAT 40
AVE=FLOAT(MEAN)/FLOAT(NTRIV)	VSTAT 41
CALL PAGER(3)	VSTAT 42
PRINT 2003, AVE,MOST,MXCDES	VSTAT 43
2003 FORMAT(17H AVERAGE LENGTH =F5.1/17H MAXIMUM LENGTH =13/	VSTAT 44
, 19H MAXIMUM POSSIBLE =13)	VSTAT 45
30 CONTINUE	VSTAT 46
IF(NVOC.LE.0) GO TO 60	VSTAT 47
C VOCABULARY WORDS	VSTAT 48
CALL PAGER(-1)	VSTAT 49
CALL PAGER(1)	VSTAT 50
PRINT 3000	VSTAT 51
3000 FORMAT(20H VOCABULARY WORDS...)	VSTAT 52
CALL MOVE(0,LENTH,-MXCDES)	VSTAT 53
DO 40 K=1,NDICT	VSTAT 54
IF(ISIGN(1,LDICT(K)).LT.0) GO TO 40	VSTAT 55
ND=IABS(LDICT(K+1))-IABS(LDICT(K))	VSTAT 56
LENTH(ND)=LENTH(ND)+1	VSTAT 57
40 CONTINUE	VSTAT 58

MOST=0	VSTAT 59
MEAN=0	VSTAT 60
DO 50 J=1,MXCODES	VSTAT 61
IF(LENTH(J).NE.0) MOST=J	VSTAT 62
50 MEAN=MEAN+J*LENTH(J)	VSTAT 63
CALL PAGER(6)	VSTAT 64
PRINT 2001, (LENTH(J),J=1,MOST)	VSTAT 65
PRINT 3002,NVOC	VSTAT 66
3002 FORMAT(31H NUMBER OF VOCABULARY ENTRIES =15)	VSTAT 67
AVE=FLOAT(MEAN)/FLOAT(NVOC)	VSTAT 68
CALL PAGER(3)	VSTAT 69
PRINT 2003, AVE,MOST,MXCODES	VSTAT 70
C OVERALL CHARACTERISTICS,	VSTAT 71
60 CONTINUE	VSTAT 72
CALL PAGER(-1)	VSTAT 73
ND=IABS(LDICT(NDICT+1))	VSTAT 74
CALL PAGER(3)	VSTAT 75
PRINT 4000,NDICT,MXENT,ND,MXDICT	VSTAT 76
4000 FORMAT(19H DICTIONARY SIZE.../20H NUMBER OF ENTRIES =14,12X,	VSTAT 77
,10H MAXIMUM =16/23H NUMBER OF CHARACTERS =16,7X,10H MAXIMUM =16)	VSTAT 76
CALL PAGER(-2)	VSTAT 79
RETURN	VSTAT 80
END	VSTAT 81

APPENDIX D
OUTLINE FOR PUBLICATION

APPENDIX D

PROGRAM MODIFICATIONS FOR PUBLICATION OF "A BIBLIOGRAPHY OF QUANTITATIVE ECOLOGY"

APPENDIX D.   Program Modifications for Publication of
"A Bibliography of Quantitative Ecology"

The following figures and text are to explain how the output from some FAMULUS main programs were changed to print material suitable for publication of the book "A Bibliography of Quantitative Ecology."

Program BRACKET used the data file BIBLI02 and inserted brackets ([and]) around the descriptor words in the DESC field in each citation. Program BRACKET consists of FAMLIB control cards using program EDIT and modifications to subroutine ADAM (Fig. D1). The file SPECIAL-BIBLIO containing brackets was cataloged as a normal EDIT tape.

Then program SEARCH, along with modifications to subroutine PRETTY, (Fig. D2) was compiled (with changes made by John R. Tadlock) to use the tape produced by the program BRACKET above, in order to divide the 4900 citations into 28 sections, based on the words in the CHK field. The output file was written 35 characters wide and stored in EDIT tape format (named PRINT).

Program TWOOLM (Fig. D3) was written to change the one column SEARCH output tape into two column format (Fig. D4). This program also functioned to insert blank lines between citations so that both columns were as even in length as possible. A few columns were one or two lines longer than others which occurred because we elected to end each page with a complete citation. In addition, TWOOLM printed headings and page numbers as well as punched a "cross reference index." This set of index cards gave us a way of "remembering" which page contained each specific citation by the citation number. Citation numbers were supplied by the main program EDIT as a unique identification for each citation (Fig. 6).

In order to produce an Author Index, main program INDEX was run on the DESA field of the alphabetically sorted EDIT tape BIBLI02. Then subroutine XCHANGE (Fig. D5) exchanged these author-citation numbers for page

numbers (punched on the output cards from program TWOCOLM), shown on the data cards at the end of Fig. D5. INDEX then produced a normal index listing (as shown in section on program INDEX). Both because a single author could be referenced more than once on the same page and the fact that page number references did not occur in order, this list was unsuitable for publication. Therefore, this list was hand corrected, keypunched and used as input to program THOMAS (Fig. D6) which listed the author names and page numbers as shown in Fig. D7.

A flowchart of the complete sequence of steps outlined above is in Fig. D8.

```

BRACKET,NT1.....TADLOCK,JR 331B,300A-23251 EXT 329
ACCOUNT(CUB438,CUB438)
ATTACH(FAMLIB,ID=FAMULUS,LC=1)
TAPESET(SPECIALBIBLIO,ID=FAMULUS,NC=10)
ATTACH(TAPE1,BIBLIO2,ID=FAMULUS,EC=K)
FTN,SYSEDT,R=3)
TAPE(TAPE2,SPECIALBIBLIO,ID=FAMULUS,W)
LDSET(LIB=FAMLIB)
LOAD,LGO.
EDIT.
REWIND(TAPE2)
SKIPF(TAPE2,1,17)
SAVTAPE(TAPE2)
RETURN(TAPE2)
AUDTAPE(ID=FAMULUS)
EXIT(S)
AUDTAPE(ID=FAMULUS)
X00  ==END OF RECORD CARD=====
      SUBROUTINE PRNCIT      (X1,X2,X3,X4)
      RETURN
      END

```

```

      SUBROUTINE ADAM(TEXT1,LCH1,NCH1,TEXT2,LCH2,NCH2,PATREP,LPAT,LREP, ADAM 2
      . NCPAT,NCREP,NPAT,MAYBE,DIFF,NAR) ADAM 3
C THIS SUBROUTINE HANDLES THE DETAILS OF THE REPLACE STATEMENT IN ADAM 4
C FAMULUS EDIT. IT OPERATES ON A SINGLE FIELD, MAKING A REPLACEMENT ADAM 5
C WHEREVER A GIVEN PATTERN MATCHES THE TEXT. ADAM 6
C THE TEXT OF THE FIELD IS THUS TRANSFERRED FROM TEXT1 SUBSTRING TO ADAM 7
C TEXT2 SUBSTRING AND THE NUMBER OF CHARACTERS IN THE LATTER IS SET AS ADAM 8
C NCH2. ADAM 9
C A PATTERN WITH ZERO NUMBER OF CHARACTERS IS OK. IT HAS THE EFFECT OF ADAM 10
C ADDING MATERIAL ONTO THE END, AND, IF THE TEXT1 FIELD IS ALSO VOID ADAM 11
C (NCH1=0), THE EFFECT IS TO CREATE A FIELD. ADAM 12
C A REPLACEMENT WITH ZERO NUMBER OF CHARACTERS DELETES TEXT, IN CASE THE ADAM 13
C CORRESPONDING PATTERN OR PATTERNS COVERS THE ENTIRE FIELD, THE EFFECT ADAM 14
C IS TO DELETE A FIELD. ADAM 15
C IN THE RESULTING TEXT2 FIELD, ANY LEADING OR TRAILING BLANKS ARE ADAM 16
C ELIMINATED. ADAM 17
C IN CASE ANY REPLACEMENT IS MADE, DIFF IS SET TO TRUE. OTHERWISE, DIFF ADAM 18
C IS UNALTERED. ADAM 19
C LCH1 POINTS TO THE START OF THE FIELD IN TEXT1. ADAM 20
C LCH1 MUST BE A MULTIPLE OF NCHW. ADAM 21
C LCH2 IS SIMILARLY DEFINED* FOR TEXT2. ADAM 22
C NCH1 IS THE NUMBER OF CHARACTERS IN THE TEXT1 FIELD TO BE PROCESSED. ADAM 23
C NCH2 IS THE NUMBER OF CHARACTERS IN THE RESULTING TEXT2 FIELD. ADAM 24
C PATREP IS A CHARACTER STRING CONTAINING THE TEXT OF PATTERNS AND ADAM 25
C REPLACEMENTS. ADAM 26
C NCPAT(J) IS THE NUMBER OF CHARACTERS IN PATTERN J. ADAM 27
C NCREP(J) IS THE NUMBER OF CHARACTERS IN REPLACEMENT J. ADAM 28
C LPAT(J) IS A POINTER TO PATREP FOR THE FIRST CHARACTER IN PATTERN J. ADAM 29
C THE CHARACTER POSITION IS LPAT(J)+1. ADAM 30
C LREP(J) IS A POINTER TO PATREP FOR THE FIRST CHARACTER IN REPLACEMENT ADAM 31
C J. THE CHARACTER POSITION IS LREP(J)+1. ADAM 32
C NPAT IS THE NUMBER OF PATTERNS (AND THE NUMBER OF REPLACEMENTS). ADAM 33
C IF NPAT=0, THE FIELD IS COPIED INTO TEXT2 WITHOUT CHANGE. ADAM 34
C NAH IS THE NUMBER OF CHARACTERS IN A LEADING ARTICLE, IF ANY, IN ADAM 35

```

FIGURE D1. Program BRACKET with Subroutine ADAM

C RESULT FIELD. (ARTICLES ARE A, AN, THE).	ADAM	36
C MAYBE(J)=1 IF CHARACTER CODE J-1 IS THE FIRST CHARACTER OF ONE OF THE	ADAM	37
C PATTERNS, =0 OTHERWISE. THIS IS FOR SPEED.	ADAM	38
COMMON /CORE/NCHW,BLANKS,BLANK	ADAM	39
INTEGER BLANKS,BLANK	ADAM	40
COMMON /FAM/SLASH,STAR,COMMA,ZERO,MXCHAR,MXCODES,MXWDES,MXCHW,	ADAM	41
MXFREE,MXWFRE	ADAM	42
INTEGER SLASH,STAR,COMMA,ZERO	ADAM	43
COMMON /TEST/ITEST	ADAM	44
INTEGER TEXT1(2),TEXT2(2),PATREP(2),LPAT(2),LREP(2),NCPAT(2),	ADAM	45
NCREP(2),MAYBE(64)	ADAM	46
LOGICAL DIFF	ADAM	47
KUT(N)=(N+NCHW-1)/NCHW	ADAM	48
C INITIALIZE.	ADAM	49
LC1=LCH1	ADAM	50
LC2=LCH2	ADAM	51
LEFT1=NCH1	ADAM	52
LEFT2=MXCHAR-LC2	ADAM	53
IF(ITEST.EQ.0) GO TO 8	ADAM	54
CALL BUTLER	ADAM	55
CALL SPOT(LC1,LEFT1,LC2,LEFT2,4HIIII,24HLC1 LEFT1 LC2 LEFT2)	ADAM	56
CALL SPOT(NPAT,DIFF,0,0,4HIIII,24HNPAT DIFF)	ADAM	57
CALL TALMUD(5HTEXT1,TEXT1,LCH1,NCH1)	ADAM	58
CALL KORAN(5HMAYBE,MAYBE,64)	ADAM	59
IF(NPAT.LE.0) GO TO 6	ADAM	60
DO 4 JP=1,NPAT	ADAM	61
CALL TALMUD(3HPAT,PATREP,LPAT(JP),NCPAT(JP))	ADAM	62
4 CALL TALMUD(3HREP,PATREP,LREP(JP),NCREP(JP))	AUAM	63
6 CALL PAGER(-1)	ADAM	64
8 CONTINUE	ADAM	65
IF(NPAT.LE.0) GO TO 90	ADAM	66
IF (LEFT1 .EQ.0) GO TO 90		
CALL CHOP("[[[[[[[[[[",1,TEXT2,LC2,1)		
LC2 = LC2 + 1		
LEFT2 = LEFT2 - 1		
DIFF = .TRUE.		
M=0	ADAM	67
C SCAN FOR START OF NEXT POSSIBLE PATTERN.	AUAM	68
10 CALL SCAN(1,MAYBE,TEXT1,LC1+M,LEFT1-M,NC,KH)	ADAM	69
MORE=MIND(NC+M,LEFT2)	ADAM	70
C COPY CHARACTERS THAT CANNOT BE FIRST CHARACTER OF A PATTERN.	AUAM	71
CALL CHOP(TEXT1,LC1,TEXT2,LC2,MORE)	ADAM	72
CALL TALMUD(5HTEXT1,TEXT1,LC1,MORE)	ADAM	73
LC1=LC1+MORE	ADAM	74
LC2=LC2+MORE	ADAM	75
LEFT1=LEFT1-MORE	ADAM	76
LEFT2=LEFT2-MORE	ADAM	77
IF(LEFT1.LE.0.OR.LEFT2.LE.0) GO TO 70	ADAM	78
C COMPARE EACH PATTERN WITH TEXT1.	ADAM	79
DO 40 JP=1,NPAT	ADAM	80
CALL SIMILE(0,PATREP,LPAT(JP),NCPAT(JP),TEXT1,LC1,LEFT1,LIKE)	ADAM	81
IF(LIKE.EQ.1) GO TO 50	ADAM	82
40 CONTINUE	ADAM	83

FIGURE D1. Program BRACKET with Subroutine ADAM
(Continued)

C NO PATTERNS MATCHED.	ADA 1	64
M=1	ADAM	85
GO TO 10	ADAM	86
C A PATTERN MATCHED. IYSEHT REPLACEMENT.	ADAM	87
50 MORE=MIN0(NCREP(JP),LEFT2)	ADAM	88
CALL CHOP(PATREP,LREP(JP),TEXT2,LC2,MORE)	ADAM	89
CALL SPOT(JP,MORE,NCPAT(JP),NCREP(JP),4HI1111,	ADAM	90
24HJP MORE NCPAT NCREP)	ADAM	91
CALL TALMUD(3HREP,PATREP,LREP(JP),MORE)	ADAM	92
CALL TALMUD(5HTEXT2,TEXT2,LC2,MORE)	ADAM	93
DIFF=.TRUE,	AWAY	94
M=0	ADAM	95
LC1=LC1+NCPAT(JP)	AUAM	96
LC2=LC2+MORE	ADAM	97
LEFT1=LEFT1-NCPAT(JP)	ADAM	98
LEFT2=LEFT2-MORE	ADAM	99
IF(LEFT1.GT.0.AND.LEFT2.GT.0) GO TO 10	ADAM	100
C INSERT MATERIAL WHICH REPLACES AN EMPTY PATTERN.	ADAM	101
70 CONTINUE.	ADAM	102
CALL SPOT(LEFT1,LEFT2,0,0,4HI1111,24HLEFT1 LEFT2	ADAM	103
DO 80 JP=1,NPAT	ADAM	104
IF(NCPAT(JP).GT.0) GO TO 80	ADAM	105
NRP=MIN0(NCREP(JP),LEFT2)	ADAM	106
CALL CHOP(PATREP,LREP(JP),TEXT2,LC2,NRP)	ADAM	107
CALL TALMUD(3HREP,PATREP,LREP(JP),NRP)	ADAM	108
CALL TALMUD(5HTEXT2,TEXT2,LC2,NRP)	ADAM	109
DIFF=.TRUE.	ADAM	110
LC2=LC2+NRP	ADAM	111
LEFT2=LEFT2-NRP	ALJAM	112
80 CONTINUE	ADAM	113
NCH2=LC2-LCH2	ADAM	114
CALL JUST(TEXT2,LCH2,NCH2,NSUB,NADD)	ADAM	115
NCH2=NCH2-NSUB	ADAM	116
GO TO 100	ADAM	117
C COPY FIELD INTACT BECAUSE NPAT=0.	ADAM	118
90 J1=KUT(LC1)+1	ADAM	119
J2=KUT(LC2)+1	ADAM	120
CALL MOVE(TEXT1(J1),TEXT2(J2),KUT(NCH1))	ADAM	121
NCH2=NCH1	ADAY	122
C RETURN	ADAM	123
100 CALL ARTY(TEXT2,LCH2,NCH2,NAR)	ADAM	124
CALL SPOT(NCH2,NAR,DIFF,0,4HI1111,24HNCH2 NAR DIFF	ADAM	125
CALL TALMUD(5HTEXT2,TEXT2,LCH2,NCH2)	ADAM	126
RETURN	ADAM	127
C TEMPORARY FORTRAN VARIABLES ARE ...	ADAM	128
C LC1 IS THE UPDATED LCH1, AS TEXT1 IS SCANNED.	ADAM	129
C LC2 IS THE UPDATED LCH2, AS TEXT2 IS FILLED.	AUAM	130
C JP IS THE INDEX OF THE PATTERN CURRENTLY BEING PROCESSED.	ADAM	131
C LEFT1 IS THE NUMBER OF CHARACTERS IN TEXT1 FIELD THAT ARE STILL TO BE	ADAM	132
C PROCESSED (SCANNED).	ADAM	133
C LEFT2 IS THE MAXIMUM NUMBER OF CHARACTERS THAT TEXT2 STILL HAS ROOM	ADAM	134
C FOR.	AUAM	135
END	ADAM	336
X00 ==END OF RECORD CARD=====		
/ID/BIBLIOGRAPHY		
/WIDTH/(047)		
/REPLACE/(1-5000)(DESC)**]*		
W 0/6/7/8/9 MOHAWK UT-200 ENJ-OF-JOB CARD		
W 0/6/7/8/9 MOHAWK UT-200 END-OF-JOB CARD		

FIGURE D1. Program BRACKET with Subroutine ADAM
(Continued)

```

SEARCH,CM60000,NT1,T400,101776,.....TABLOCK,JOHN,,23251,,,331-300 AREA
ACCOUNT(CUR438,CUB438)
ATTACH(FAMLIB,ID=FAMULUS).
TAPESET PRINTFILEOFSEARCH,ID=FAMULUS,NC=10)
FTN(SYSEDT,B=SUB).
SUMMARY.
TAPE XDATA,SPECIALBIBLIO,ID=FAMULUS)
REQUEST(DATA,*EC)
REWIND(XDATA)
COPYBF(XDATA,DATA).
UNLOAD(XDATA)
SUMMARY.
TAPE(PRINT,PRINTFILEOFSEARCH,ID=FAMULUS,W)
SUMMARY.
LOAD(SUB)
LDSET(MAP=0,LIR=FAMLIB).
SEARCH(,PRINT,DATA)
SUMMARY.
RETURN(DATA)
SAVTAPE(PRINT)
SUMMARY.
REWIND(PRINT)
COMMENT,..NEXT CARD WILL CAUSE 400 LINES OF FILE TO LIST..THEN COPY TERMINATES.
COPYSF(PRINT,,C,LC=400)
JOBNAME(WPR,PW)
REWIND(PRINT)
COPYCF(PRINT,WPR)
JOBNAME(WPR,EJ,3)
ROUTE(WPR,DC=PR,TID=27)
EXIT(S)
EXIT(S)
X
      SUBROUTINE PAGER (N)
      IF ( N.EQ. 1) RETURN
      PRINT 7
7      FORMAT("                ")
      RETURN
      END
      SUBROUTINE PRETTY(CIT,LCIT,INDENT,LEN,FCHOI,JLAB,NCHOI)
C PRINTS A CITATION IN BIBLIOGRAPHIC FORMAT.
C THE CITATION IS AT CIT.
C THE CITATION NUMBER IS LCIT, WHICH IS PRINTED ON THE LEFT UNLESS IT IS
C 0 IN WHICH CASE IT IS NOT PRINTED.
C INDENT=1 SAYS TO INDENT ALL LINES BUT THE FIRST.
C INDENT=0 SAYS TO INDENT NONE OF THEM.
C THE FIELDS ARE PRINTED IN THE ORDER DICTATED BY THE IFIELD ARRAY.
C LEN IS THE PAGE WIDTH, IN CHARACTERS. IT IS THE MAX. NUMBER OF
C CITATION TEXT CHAHACTERS ON A LINE PLUS 4 CHARACTERS FOR LCIT(IF LCIT
C IS NOT 0).
      COMMON /CORE/NCHW,BLANKS,BLANK
      INTEGER BLANKS,BLANK
      COMMON /FIEL/FIELD(10),IFIELD(10),NFIELD,KDESCR
      INTEGER FIELD
      INTEGER CIT(3,2),LINE(16)
C THE DIMENSION OF LINE() IS LENLIN+1.
      LOGICAL GOLINE,GOCHAR,ENTROP,BLOCK,DONE
      LOGICAL FCHOI

```

```

07/18/75
PRETTY 3
PRETTY 4
PRETTY 5
PRETTY 6
PRETTY 7
PRETTY 8
PRETTY 9
PRETTY10
PRETTY11
PRETTY12
PRETTY13
PRETTY14
PRETTY15
PRETTY16
PRETTY17
PRETTY18
PRETTY19
07/18/76

```

FIGURE D2. SEARCH Control Cards and Program Modification to Produce a Data File

INTEGER JLAB(10)	07/18/77
DATA LENLIN/15/	PRETTY20
C INITIALIZE	PRETTY21
GO LINE=.FALSE.	PRETTY22
KF=0	PRETTY23
NC=0	PRETTY24
LL=0	PRETTY25
ENTROP=LCIT,EQ.0	PRETTY26
BLOCK=INDENT.LE.0	PRETTY27
MOST=MIN0(LEN-7,LENLIN*NCHW)	PRETTY28
IF(ENTROP) MOST=LEN	PRETTY29
DONE=.FALSE.	PRETTY30
CCCC CALL PAGER(-1)	
PRINT 9673	
9673 FORMAT ("+")	
C PREPARE TO START A NEW LINE.	PRETTY32
in CALL MOVE(BLANKS,LINE,-LENLIN)	PRETTY33
GOCHAR=.FALSE.	PRETTY34
NL=MOST-LL	PRETTY35
C TEST IF CURRENT FIELD DEPLETED AND IF SO, GET NEXT FIELD.	PRETTY36
20 IF(NC.GT.0) GO TO 30	PRETTY37
IF(FCHOI) GO TO 25	07/18/78
IF(KF,GE,NFIELD) GO TO 80	PRETTY38
KF=KF+1	PRETTY39
KFF=IFIELD(KF)	PRETTY40
GO TO 27	07/18/79
25 IF(KF,GE,NCHOI) GO TO 80	07/18/10
KF=KF+1	07/18/11
KFF=JLAB(KF)	07/18/12
27 NC=CIT(1,KFF)	07/18/13
LC=CIT(2,KFF)	PRETTY42
IF(.NOT.(GOCHAR.AND.NC.GT.0)) GO TO 20	PRETTY43
N=MIN0(NL,3)	PRETTY44
CALL CHOP(BLANKS,0,LINE,LL,-N)	PRETTY45
LL=LL+N	PRETTY46
NL=NL-N	PRETTY47
GO TO 20	PRETTY48
C SELECT GROUP OF CHARS TO MOVE INTO LINE WAITING TO PRINT.	PRETTY49
30 CALL SCOUT(0,0,BLANK,CIT,LC,NC,M)	PRETTY50
LC=LC+M	PRETTY51
NC=NC-M	PRETTY52
N=MIN0(NC,NL+1)	PRETTY53
IF(NC.LE.NL) GO TO 35	PRETTY54
CALL SCOUT(1,1,BLANK,CIT,LC,N,M)	PRETTY55
IF(M.EQ.N.AND..NOT.GOCHAR) M=1	PRETTY56
N=N-M	PRETTY57
35 IF(N.LE.0) GO TO 40	PRETTY58
CALL CHOP(CIT,LC,LINE,LL,N)	PRETTY59
LC=LC+N	PRETTY60
NC=NC-N	PRETTY61
LL=LL+N	PRETTY62
NL=NL-N	PRETTY63
GOCHAR=.TRUE.	PRETTY64
GO TO 20	PRETTY65
C PRINT THE LINE WAITING.	PRETTY66
C- 40 CALL PAGER(1)	PRETTY67
40 CONTINUE	

FIGURE D2. SEARCH Control Cards and Program Modification
to Produce a Data File (Continued)

IF(ENTROP) GO TO 60	PRETTY68
IF(GOLINE) GO TO 50	PRETTY69
C---PRINT FIRST LINE WITH CITATION NUMBER	07/18/14
GOLINE=.TRUE.	PRETTY70
PRINT 1000,LCIT,(LINE(J),J=1,LENLIN)	PRETTY71
C	
C-----SPECIAL FORMAT WITH NUMBFRS FOR GREEN BOOK - 12/3/75/M COCHRAN	
C	
1000 FORMAT (4X,I4,".",3X,15A8)	
C1000 FORMAT (2X,I4,".",1X,15A6)	
C	
GO TO 70	PRETTY73
C---PRINT REMAINING LINES UNDER COMMON FORMAT	07/18/17
50 PRINT 2000, (LINE(J),J=1,LENLIN)	PRETTY74
2000 FORMAT (12X,15A8)	07/18/18
GO TO 70	PRETTY76
C---PRINT LINE WITHOUT CITATION NUMBER WITH FIRST LINE OFFSET - RJ OLSON	07/18/19
60 PRINT 3000, (LINE(J),J=1,LENLIN)	PRETTY77
ENTROP = .FALSE.	07/18/20
GOLINE = .TRUE.	07/18/21
3000 FORMAT(8X,15A8)	07/18/22
C RESET FOR NEXT LINE, IF ANY.	PRETTY79
70 IF(DONE) GO TO 100	PRETTY80
LL=5	PRETTY81
IF(BLOCK) LL=0	PRETTY82
GO TO 10	PRETTY83
C FINISHED WITH CITATION, PUT MUST PRINT PARTIAL LAST LINE.	PRETTY84
DO DONE=.TRUE.	PHETTY85
GO 10 40	PRETTY86
100 CONTINUE	
PRINT 9667	
9667 FORMAT (" ")	
C ALL DONE. RETURN,	PRETTY87
RETURN	PRETTY88
END	PRETTY89
X	
/ID/BIBLIOGRAPHY	
/FIELD CHOICE/(AUTH,YEAR,TITL,REF,DESC)	
/WIDTH/(42)	
/SELECT/(1-5555)	
/FIELDS/(CHK)	
/SEARCH/(AGE)	
/SEARCH/(COMP-MODEL)	
/SEARCH/(COMPET-DEN)	
/SEARCH/(COMPUTERS)	
/SEARCH/(SIMULAT)	
/SEARCH/(CYCLES)	
/SEARCH/(DIVERSITY)	
/SEARCH/(ENERGET)	
/SEARCH/(EXPL-MNG)	
/SEARCH/(FREQ-DIST)	
/SEARCH/(GROW-IND)	
/SEARCH/(POPUL-GRO)	
/SEARCH/(MATH-MODEL)	
/SEARCH/(POPUL-MODEL)	
/SEARCH/(STOC-MODEL)	
/SEARCH/(OPER-RES)	

FIGURE D2. SEARCH Control Cards and Program Modification to Produce a Data File (Continued)

```

/SEARCH/(ORDINA)
/SEARCH/(PATTERNS)
/SEARCH/(PLOTLESS)
/SEARCH/(PLOT-QUA)
/SEARCH/(POPUL-DYNA)
/SEARCH/(CAPT-RECAPT)
/SEARCH/(POPUL-EST)
/SEARCH/(REPRODUCT)
/SEARCH/(SAMPL)
/SEARCH/(SURV-MORT)
/SEARCH/(TAXON)
/SEARCH/(BOOK)
/SEARCH/(BIBLIOGRAPHY)
/SEARCH/(EDUCATION)
W      0/6/7/8/9      MOHAWK  UT-200  END-OF-JOB  CARD
W      0/6/7/8/9      MOHAWK  UT-200  END-OF-JOB  CARD

```

FIGURE 02. SEARCH Control Cards and Program Modification
to Produce a Data File (continued)

```

COLM2,NT1,CM52400,10200,T60.....TAULOCK, JOAN 331BLDG,300 AREA 23251,...
ACCOUNT(CUB438,CUB438)
ROUTE(FTNMAP,DEF,DC=PR,TID=27,FID=*),
JOBNAME(FTNMAP,3)
FTN(A,OPT=2,R=3,P,L=FTNMAP)
PRNTLMT(33333)
JOBNAME(PUNCH)
ROUTE(PUNCH,DEF,FID=*,TID=27)
TAPE(PRINT,PRINTFILEOFSEARCH,ID=FAMULUS,R,CY=HC-1)
SUMMARY,
LDSET(MAP=/FTNMAP),
LGO.
SUMMARY,
ROUTE(OUTPUT,DEF,DC=PR,TID=27,FC=LL).....ECOLOGY SPECIAL PRODUCTION
JOBNAME(FTNMAP,3,CN=$END OF JOBS)
SUMMARY,
REWIND(PUNCH)
COPYSE(PUNCH,,LR,EOL)
EXIT,S.
JOBNAME(FTNMAP,3,CN=$END OF JOBS)
EXIT,S.
EXIT(S)
EXIT(S)
TAPE(PRINT,PRINTFILEOFSEARCH,ID=FAMULUS,R)
X
PROGRAM TWOCOLM (OUTPUT,PUNCH,FTNMAP,TAPE8=FTNMAP)
LOGICAL GARBAGE
EXTERNAL EOFCALL
COMMON /XFILEX/ FITABLE(35),IEND
COMMON // KROSS (5000)
PRINT 200
DO 50 I=1,5000
50 KROSS(I)=-0
CALL FILESQ (FITABLE,3LLFN,5LPRINT,2LRT,1LZ,2LBT,1LC,2LFL,138,2LDX
*,EOFCALL,3LERL,1,3LSDS,3LYES,2LCM,3LYES)
CALL OPENM (FITABLE,5LINPUT,1LR)
IEND=0
CALL SETPAGE
GARBAGE=,TRUE,
C
C THE ALTERNATE RETURNS FROM GETCIT ARE USED WHEN
C 1. ENC OF FILE IS READ IN THE EXPECTED PLACE
C 2. MISC. LINES OF THE PRINT FILE ARE READ (NOT CITATIONS)
C 3 AN ERROR CONDITION IS ENCOUNTERED ( TERMINATE JOB )
100 CALL GETCIT, RETURNS (130,110,120)
GARBAGE=,FALSE,
CALL ADPPAGE
GO TO 100
110 IF (GARBAGE) GO TO 100
GARBAGE=,TRUE,
CALL OUTPAGE
GO TO 100
120 CONTINUE
CALL REMARK (" ERROR IN PROGRAM ,")
130 CALL OUTPAGE
CALL CLOSEM (FITABLE,1LU)
PRINT 200

```

	A	1
	A	2
	A	3
	A	4
	A	5
	A	6
	A	7
	A	8
	A	9
	A	10
	A	11
	A	12
	A	13
	A	14
	A	15
	A	16
	A	17
	A	18
	A	19
	A	20
	A	21
	4	22
	A	23
	A	24
	A	25
	A	26
	A	27

FIGURE D3. Program TWOCOLM to Create Two Column Format

	CALL PRITITLE	A	28
	STOP "-----"	A	29
200	FORMAT ("1")		
	END	A	30
	SUBROUTINE GETCIT, RETURNS (MR1,MR2,MR3)	B	1
C	-----ROUTINE TO READ A CITATION FROM A PRINT FILE CREATED BY FAM	B	2
C	-----NUMBER WILL CONTAIN THE FAMULUS CITATIOY NUMBER	B	3
C	----- IIRAY WILL CONTAIN THE CITATION	B	4
C	----- NLINE WILL BE THE LENGTH	B	5
C	----- RETURN MR1 IF END OF FILE IS READ IN THE NORMAL PLACE	B	6
C	----- RETURN MR2 IF A NEW PAGE IS NEEDED ,NEW SECTION	B	7
C	----- RETURN HR3 IF AN ABNORMAL CONDITION OCCURS AND MUST ABORT	B	8
C	----- RETURN NORMAL RETURN WITH A CITATION	B	9
	COMMON /XHEADX/ LONGHED,IPAGEND,IGROUP	B	10
	COMMON /XFILEX/ FITABLE(35),IEND	B	11
	DIMENSION LINE(14), IIRAY(120), KOUT(800)	B	12
	EQUIVALENCE (IIRAY,IIRAY), (OUT,KOUT)	B	13
	*, (LT1,LTITLE(1,1)), (LT2,LTITLE(1,9)), (LT3,LTITLE(1,17	B	14
	+)), (LT4,LTITLE(1,25))	B	15
	INTEGER LT1(72),LT2(72),LT3(72),LT4(72)	B	16
	COMMON /XPAGEX/ IPAGE,IHEAD(5),LTITLE(9,32),OUT(4,200),NUMBER,NLIN	B	17
	+E,IIRAY(4,30)	B	18
C		B	19

	DATA LT1 /
	+ "H/(AGE) ", 03, 1, 0,
	+ "AGE' ", 4* " ",
	+ "H/(COMP-MO", 18, 1, 0,
	+ "COMPARTMEN", "T MODELS ", 3* " ",
	+ "H/(COMPET-", 47, 1, 0,
	+ "COMPETITIO", "N, DENSITY", "-DEPENDENC", "E, AND REG", "ULATION ",
	+ "H/(COMPUTE", 29, 1, 0,
	+ "COMPUTERS ", "AND DATA-P", "ROCESSING ", 2* " ",
	+ "H/(SIMULAT", 19, 1, 0,
	+ "COMPUTER S", "IMULATION ", 3* " ",
	+ "H/(CYCLES)", 06, 1, 0,
	+ "CYCLES ", 4* " ",
	+ "H/(DIVERSI", 09, 1, 0,
	+ "DIVERSITY ", 4* " ",
	+ "H/(ENERGET", 27, 1, 0,
	+ "ENERGETICS", " AND PRODU", "CTIVITY ", 2* " "/

	DATA LT2 /
	+ "H/(EXPL-MN", 37, 1, 0,
	+ "EXPLOITATI", "ON, MANAGE", "MENT, AND ", "CONTROL ", "
	+ "H/(FREQ-DI", 23, 1, 0,
	+ "FREQUENCY ", "DISTRIBUTI", "ONS ", " ", " ", "
	+ "H/(GROW-IN", 21, 1, 0,
	+ "GROWTH OF ", "INDIVIDUAL", "S ", " ", " ", "
	+ "H/(POPUL-G", 21, 11 0,
	+ "GROWTH CF ", "POPULATION", "S ", " ", " ", "
	+ "H/(MATH-MO", 21, 1, 0,
	+ "MODELS (MA", "THEMATICAL", " ", " ", " ", "
	+ "H/(POPUL-M", 19, 1, 0,
	+ "MODELS (PO", "PULATION) ", " ", " ", "
	+ "H/(STOC-MO", 19, 1, 0,
	+ "MODELS (ST", "OCHASTIC) ", " ", " ", "
	+ "H/(OPER-RE", 35, 1, 0,

FIGURE D3. Program TWOCOLM to Create Two Column Format (Continued)

	IF (LINE(3),EQ,LTITLE(1,I)) GO TO 110	B	78
100	CONTINUE	B	79
	CALL REMARK (" NO MATCH ON SEARCH CARD")	B	80
	RETURN MP3	B	81
C	-----		
110	DO 120 I=1,5	B	82
120	IHEAD(I)=LTITLE(I+4,ISAVE)	B	83
	LONGHED=LTITLE(2,ISAVE)	B	84
	IGROUP=ISAVE	B	85
	IPAGEND=LTITLE(3,ISAVE)	B	86
	LTITLE(3,ISAVE)=IPAGE		
	IF (LINE(3),NE, "H/(EDUCATI") GO TO 122		
	NT1="EDUCATIONA" \$ NT2="L REFERENC" \$ NT3="ES "		
	NUMBER=" 4803,"		
	GO TO 130		
122	IF (LINE(3),NE, "H/(RIBLIOG") GO TO 124		
	NT1="BIBLIOGRAP" \$ NT2="HIES" \$ NT3=" "		
	NUMBER=" 4802,"		
	GO TO 130		
C	-----		
C	WRITE A PAGE BETWEEN EACH SECTION	B	88
124	CALL HEACERA		
	LTITLE(3,ISAVE)=IPAGE - 1		
	IF (LINE(3),NE, "H/(BOOK) ") RETURN MR2		
C	-----		
	NT1="BOOKS" \$ NT2=NT3=" "		
	NUMBER=" 4801,"		
130	DO 136 I=1,120		
136	IIRAY(I) = " "		
	IRAY(2,4)=NT1\$IRAY(3,4)=NT2 \$ IRAY(4,4)= NT3\$ IRAY(1,4)=" ,,"		
	NLINE = 6		
	RETURN		
C	-----	B	93
140	DO 150 I=1,120	B	95
C	----- CLEAR OUT CITATION ARRAY	B	96
150	IIRAY(I)=" "	B	97
160	CONTINUE	B	98
	CALL GET (FITABLE,LINE)	B	99
	IF (IEND,NE,0) CALL REMARK ("-----EOF IN WRONG PLACE-----")	B	100
	IF (IEND,NE,0) RETURN MR3	B	101
C	----- CHECK IF THIS IS A BLANK LINE THAT FOLLOWS EACH CITATION	B	102
	IF (LINE(2),EQ," ") GO TO 160	B	103
	IF (LINE(1),NE," ") NUMBER=LINE(1)	B	104
	IF (NLINE,GE,30) GO TO 190	B	105
	NLINE=NLINE+1	B	106
	DO 170 I=1,4	B	107
170	IRAY(I,NLINE)=LINE(I+1)	B	108
	GO TO 160	B	109
C	-----	B	110
180	CONTINUE	B	111
	IF (NUMBER,EQ," ") CALL REMARK ("-----INVALID CITATION NUMBER*)	B	112
	IF (NUMBER,EQ," ") RETURN MR3	B	113
	IF (NUMBER,EQ," 4327,") NLINE = NLINE + 8		
	IF (NUMBER,EQ," 1860,") NLINE= NLINE - 1		
	IF (NUMBER,EQ," 1634,") NLINE= NLINE - 1		
	IF (NUMBER,EQ," 2728,") NLINE: NLINE + 1		
	IF (NUMBER,EQ," 2727,") NLINE= NLINE + 1		

FIGURE D3. Program TWOOLM to Create Two Column Format (Continued)

```

+ "OPERATIONS"," RESEARCH ","AND PROGRA","MMING      ","      "/"
      DATA LT3 /
+ "H/(ORDINA)", 10, 1, 0,      ","      ","      ",
+ "ORDINATION", 08, 1, 01,      ","      ","      ",
+ "H/(PATTERN", 08, 1, 01,      ","      ","      ",
+ "PATTERNS  ",      ","      ","      ",
+ "H/(PLOTLES", 17, 1, 01,      ","      ","      ",
+ "PLOTLESS S",AMPLING  ",      ","      ",
+ "H/(PLOT-QU", 181 11 01,      ","      ","      ",
+ "PLOTS AND ", "QUADRATS  "  n, a      ","      ",
+ "H/(POPUL-D", 19, 11 0,      ","      ","      ",
+ "POPULATION", " DYNAMICS  ",      ","      ",
+ "H/(CAPT-RE", 491 1, 0,      ","      ","      ",
+ "POPULATION", " ESTIMATION", "N (CAPTURE", "-RECAPTURE", " METHODS) ",
+ "H/(POPUL-E", 371 1, 0,      ","      ","      ",
+ "POPULATION", " ESTIMATION", "N (OTHER M", "ETHODS)  ",      ",
+ "H/(REPRODU", 12, 11 0,      ","      ","      ",
+ "REPRODUCTI", "ON      ","      ","      "/"

      DATA LT4 /
+ "H/(SAMPL) ", 08, 11 0,      ","      ","      ",
+ "SAMPLING  ",      ","      ","      ",
+ "H/(SURV-MO", 22, 1, 0,      ","      ","      ",
+ "SURVIVAL A", "ND MORTALI", "TY      ","      ",
+ "H/(TAXON) ", 08, 11 01,      ","      ","      ",
+ "TAXONOMY  ",      ","      ","      ",
+ "H/(BOOK)  ", 49, 1, -1,      ","      ","      ",
+ "BOOKS, BIB", "LIOGRAPHIE", "S, AND EDU", "CATIONAL R", "EFERENCES ",
+ "H/(BIBLIOG", 49, 11 -1,      ","      ","      ",
+ "BOOKS, BIB", "LIOGRAPHIE", "S, AND EDU", "CATIONAL R", "EFERENCES ",
+ "H/(EDUCATI", 49, 1 -1,      ","      ","      ",
+ "BOOKS, BIB", "LIOGRAPHIE", "S, AND EDU", "CATIONAL R", "EFERENCES ",
+ "      ", 0 1 0 1 0 1      ","      ","      ",
+ "      ",      ","      ","      ",
+ "      ", 0 1 0 1 0 1      ","      ","      ",
+ "      ",      ","      ","      "/"

80 NLINE=0      B 65
   NUMBER=" "      B 66
C   ----- READ A LINE FROM THE FILE      B 67
   CALL GET (FITABLE,LINE)      B 68
C   --- READING END OF FILE IN NORMAL PLACE,, ,      B 69
   IF (IEND,NE,0) RETUPN MR1      B 70
C   ----- CHECK IF THIS IS A SPECIAL LINE THAT PRECEDES  EACH CITATION      B 71
   IF (LINE(1),EQ,"+") GO TO 140      B 72
300 FORMAT ("0 PAGE",I6,5X,11A10,"*",3A1)
   IF (LINE(2),NE, " ") WRITE (8,300) IPAGE,LINE
   IF (LINE(2),EQ,"D.../SEARC") GO TO 93      B 74
C   ----- READ EXTRA LINES OF PRINT FILE      B 73
C   LAST THREE GROUPS ARE TREATED DIFFERENT
   IF (IHEAD(1),EQ, "BOOKS, BIB" ) GO TO 80
   RETURN MR2
C   -----
90 DO 100 I=1,32      B 76
C   FOUND SEARCH CARD ■ NOW LOOK UP THE TITTLE IT GOES WITH      B 75
   ISAVE=I      B 77

```

FIGURE D3. Program TWOCOLM to Create Two Column Format
(Continued)

C	-----	B	114
	IF (NLINE,GT,0) RETURN	B	115
C	-----	B	116
190	CALL REMAHK ("-----INVALIDNUMBER OF LINES")	B	117
	RETURN MR3	B	118
	END	B	119
	SUBROUTINE SETPAGE	C	1
	EQUIVALENCE (KOUT,OUT), (IRAY,IIRAY)	C	2
	COMMON / / KROSS(5000)	C	3
	COMMON /XHEADX/ LONGHED,IPAGEND,IGROUP	C	4
	COMMON /XPAGEX/ IPAGE,IHEAD(5),LTITLE(9,32),OUT(4,200),NUMBER,NLIN	C	5
	+E,IRAY(4,30)	C	6
	INTEGER KOUT(800),OUT,IIRAY(120)	C	7
	LOGICAL HOLD,ENDPAGE,SPECIAL	C	8
	DATA KHALF,MAXLINE/63,125/	C	10
	HOLD=,FALSE,	C	11
100	DO 110 I=1,800	C	12
110	KOUT(I)=" "	C	13
	KOUNTBR=KOUNTBL=3	C	14
	JTEST=0	C	15
	SPECIAL=,FALSE,	C	16
	JJTEST=0	C	17
	LINECNT=0	C	18
	IF (HOLD) GO TO 120	C	19
	RETURN	C	20
C	-----	C	21
	ENTRY ADPPAGE	C	22
C	-----	C	23
120	HOLD=,TRUE,	C	24
	IF (SPECIAL) GO TO 180	C	25
	ENDPAGE=,FALSE,	C	26
	IF (LINECNT,GT,KHALF) GO TO 130	C	27
	LBLANKS=KHALF-LINECNT	C	28
	KOUNTBL=KOUNTBL+1	C	29
	GO TO 150	C	30
130	CONTINUE	C	31
	JTEST=MAXLINE-LINECNT-NLINE	C	32
	IF (JTEST,LT,0) GO TO 190	C	33
	IF (JTEST,GE,LBLANKS) GO TO 140	C	34
C	-C-CCIF (NLINE ,LE. 6) GO TO 600	C	35
C	-C-C- GO TO 6P0	C	36
C	---- DECIDE SPECIAL CASE,....	C	37
	IF ((KOUNTBL,GE,LBLANKS),AND,(KOUNTBR,GE,JJTEST-LBLANKS)) GO TO 19	C	38
	+0	C	39
	SPECIAL=,TRUE,	C	40
	WRITE(8,300) NUMBER,IPAGE	C	41
140	CONTINUE	C	42
	KOUNTBR=KOUNTBR+1	C	43
150	CONTINUE	C	44
	JJTEST=JTEST	C	45
C	-	C	46
C	DO 500 I = 1 , LONGCIT	C	47
C	DO 500 J = 1 , 4	C	48
	OUT (J,I+LINECNT) = IIRAY (J,I)	C	49
	IWORD=LINECNT*4	C	50
	LONG=NLINE*4	C	51
	DO 160 I=1,LONG	C	51
160	KOUT(IWORD+I)=IIRAY(I)	C	52

FIGURE D3. Program TWOCOLM to Create Two Column Format
(Continued)

	DECODE (9,310,NUMBER) INDEX	C	53
	IF (INDEX,LE,0) STOP " PROBLEM IN PAGE,, INDEX ,LE. 0"	C	54
	IF (KROSS(INDEX),NE,0) WRITE(8,320) NUMBER,IPAGE,KROSS(INDEX)	C	55
	KROSS(INDEX)=IPAGE	C	56
C	---- COUNT CITATIUNS IN EACH GROUP	C	57
	LTITLE(4,IGROUP)=LTITLE(4,IGROUP)+1	C	58
	LINECNT=LINECNT+1+NLINE	C	59
	HOLD=,FALSE,	C	60
	RETURN	C	61
C	-----	C	62
	ENTRY OUTPAGE	C	63
C	-----	C	64
	HOLD=,FALSE,	C	65
	IF (LINECNT,EQ,0) GO TO 100	C	66
	IF (SPECIAL) GO TO 180	C	67
	ENDPAGE=LINECNT,LT,100	C	68
	IF (,NOT,ENDPAGE) GO TO 190	C	69
	IHALF=JHALF=((LINECNT-1)/2)	C	70
	DO 170 I=1,20	C	71
	IHALF=JHALF+I	C	72
	IF (IHALF,GT,63) GO TO 190	C	73
	IF (OUT(1,IHALF),EQ," ") GO TO 280	C	74
170	CONTINUE	C	75
	GO TO 190	C	76
180	CONTINUE	C	77
C	---- SPECIAL CASE HAVE TO BREAK CITATION	C	76
	LBLANKS=0	C	79
	IF (KOUNTBR,GE,JJTEST) GO TO 190	C	80
	LBLANKS=JJTEST-KOUNTBR	C	81
190	CONTINUE	C	82
C	ADJUST LEFT COLUMN	C	83
	IF (LBLANKS,LE,0) GO TO 230	C	84
	IPOINT=1	C	85
	KLSAVE=KHALF	C	86
	DO 220 INDEX=1,LBLANKS	C	87
200	IPOINT=IPOINT+1	C	88
	IF (IPOINT,GT,KHALF) GO TO 210	C	89
	IF (OUT(1,IPOINT),NE," ") GO TO 200	C	90
	KLSAVE=IPOINT	C	91
210	CALL MOVER (OUT,KLSAVE)	C	92
	IPOINT=IPOINT+1	C	93
220	CONTINUE	C	94
230	CONTINUE	C	95
	JJTEST=JJTEST-LBLANKS	C	96
	IF (JJTEST,LE,0) GO TO 270	C	97
	KRSAVE=MAXLINE	C	98
	IPOINT=KHALF+1	C	99
	DO 260 INDEX=1,JJTEST	C	100
240	IPOINT=IPOINT+1	C	101
	IF (IPOINT,GT,MAXLINE) GO TO 250	C	102
	IF (OUT(1,IPOINT),NE," ") GO TO 240	C	103
	KRSAVE=IPOINT	C	104
250	CALL MOVER (OUT,KRSAVE)	C	105
	IPOINT=IPOINT+1	C	106
260	CONTINUE	C	107
270	CONTINUE	C	108
	IHALF=KHALF	C	109

FIGURE D3. Program TWOCOLM to Create Two Column Format
(Continued)

280	CONTINUE	C 110
	CALL HEADERX	C 111
	DO 290 INDEX=1,IHALF	C 112
290	PRINT 330, (OUT(J,INDEX),J=1,4),(OUT(J,INDEX+IHALF),J=1,4)	C 113
	GO TO 100	C 114
C		C 115
300	FORMAT ("0 PROBLEMS WITH CITATION ",A10," ON PAGE ",I4/ 60(" *"))	C 116
310	FORMAT (4X,I4,1X)	C 117
320	FORMAT (" CITATION LISTED TWICE ",A10,I6,I6)	C 118
330	FORMAT (13X,3A10,A8,4A10)	C 119
	END	C 120
	SUBROUTINE HEADERX	D 1
	COMMON /XPAGEX/ IPAGE,IHEAD(5),LTITLE(9,32),OUT(4,200),NUMBER,NLIN	D 2
	+E,IRAY(4,30)	D 3
	COMMON / / KROSS(5000)	
	COMMON /XHEADX/ LONGHED,IPAGEND,IGROUP	D 5
	COMMON /XF ILEX/ FITABLE(35),IEND	D 6
	DATA IPAGE/77/	
C	-----EVEN ODD PAGES USE DIFFERENT HEADING SO PAGE NUM IS OUTS	D 7
	ITAB1=86	D 8
C	----- PRINT EVEN NUMBERED PAGE	D 9
	ITAB2=16	D 10
	IPTEST=IPAGE,AND,18	D 11
	IF (IPTEST,NE,0) SO TO 100	D 12
C	----- PRINT ODD NUMBERED PAGE	D 13
	ITAB2=89-LONGHED	D 14
	ITAB1=16	D 15
	IF (IPAGE,GE,100) GO TO 100	D 16
	ITAB1=15	D 17
	IF (IPAGE,GE,10) GO TO 100	D 18
	ITAB1=14	D 19
100	CONTINUE	D 20
	PRINT 120, ITAB1,IPAGE,ITAB2,IHEAD	D 21
	IPAGE=IPAGE+1	D 23
	RETURN	D 24
C	-----	
	ENTRY PRITITLE	D 25
C	-----	
	WRITE (8,180)	
	WRITE(8,130) LTITLE	D 26
	PUNCH 130, LTITLE	
	WRITE (8,180)	
	DO 110 J=1,5000,25	D 27
110	PUNCH 140, J,(KROSS(I+J-1),I=1,25)	D 28
	ITOTAL = 0	
	DO 115 I= 1,4800	
	IF(KROSS(I),NE, 0) ITOTAL = ITOTAL + 1	
	IF ((I-ITOTAL),GT, 50) GO TO 115	
	IF (KROSS(I),EQ, 0) WRITE (8,400) I	
115	CONTINUE	
	WRITE (8,500) ITOTAL	
	RETURN	D 29
C	-----	
	ENTRY EOFCALL	D 30
C	-----	
	IEND=IFETCH(FITABLE,2LFP)	D 31
	WRITE (8,150) IEND	D 32

FIGURE D3. Program TWOCCUM to Create Two Column Format
(Continued)

```

C      RETURN
C      ENTRY HEADERS
C      CHANGE NEXT PAGE TO PRINT TO 000 FOR THE TITLE PAGES
C      IF((IPAGE ,AND, 18) .EQ. 0 ) IPAGE = IPAGE + 1
C      PRINT 120, 86, IPAGE + 16 , IHEAD
C      PRINT 200, IPAGE
C      IPAGE = IPAGE + 1
C      RETURN
C      120 FORMAT ("1",T=,I3,T=,5A10/"0")
C      130 FORMAT ("0 ",A10,3I4,3X,5A10)
C      140 FORMAT (I5,25I3)
C      150 FORMAT("0"/"0"/"0" END OF FILE READ "020//")
C      160 FORMAT ("1 ")
C      200 FORMAT ( 30("0"/),T53,I3)
C      400 FORMAT (" CITATION NOT LISTED",I6)
C      500 FORMAT ("0 NUMBER OF CITATIONS PROCESSED=",I6)
C      END
C      SUBROUTINE MOVER (KARRAY,LINE)
C      INTEGER KARHAY(800)
C      - ROUTINE Ti) A33 A BLANK LINE
C      DATA LIMIT/600/
C      IF (LINE.LE.1) STOP "ERROR IN MOVER., LOW "
C      IF (LINE.GT.135) STOP "ERROR IN MOVER., HIGH"
C      LPOINT=LIMIT
C      LSTOP=LINE*4
C      IF (KARRAY(LSTOP-3).NE," ") STOP " ERROR IN MOVER "
C      100 KARRAY(LPOINT)=KARRAY(LPOINT-4)
C      LPOINT=LPOINT-1
C      IF (LPOINT.GT.LSTOP) GO TO 100
C      RETURN
C      - - - - -
C      ENTRY MOVERB
C      - - - - -
C      ISTART = (LINE*4) - 3
C      IF ( KARRAY(ISTART) .NE. " " ) STOP "ERROR IN MOVER,B....."
C      DO 500 I = ISTART,LIMIT
C      500 KARRAY(I) = KARRAY(I+4)
C      RETURN
C      END
C      F 14
W      0/6/7/8/9      MOHAWK UT-200 END-OF-JOB CARD
W      0/6/7/8/9      MOHAWK UT-200 END-OF-JOB CARD
W      H              %B 8 : 775877 07 R = +7-707177

```

FIGURE D3. Program TWOOLM to Create Two Column Format
(Continued)

- ALLEN, K. R. 1966, DETERMINATION OF AGE DISTRIBUTION FROM AGE • LENGTH KEYS AND LENGTH DISTRIBUTIONS, **IBM 7090**, 7094 FORTRAN IV, TRANS. AMER. FISH, SOC. 95(2), 230-231.
- ALLEN, R. L. AND P. BASASIBWAKI, 1974, PROPERTIES OF AGE STRUCTURE MODELS FOR FISH POPULATIONS. J, FISH, RES. BOARD CAN. 31(6), 1119-1125. [SIMULATION; STOCK-RECRUITMENT; STABILITY; RICKER'S MODEL3]
- ANDERSEN, F. S. 1960, COMPETITION IN POPULATIONS CONSISTING OF ONE AGE GROUP, BIOMETRICS 16(1), 19-27.
- BARTLETT, M. S. 1970, AGE DISTRIBUTIONS, BIOMETRICS 26(3), 377-385.
- CAUGHLEY, G. 1974, INTERPRETATION OF AGE RATIOS, J. WILDL. MANAGE. 38(3), 557-562. [COMPUTER; SIMULATION; MODEL]
- CAUGHLEY, G. AND J. CAUGHLEY, 1974, ESTIMATING MEDIAN DATE OF BIRTH, J. WILDL. MANAGE. 38(3), 552-556. [PROBIT ANALYSIS]
- COUTINHO, A. B. AND F. A. COUTINHO, 1968, THE CONSERVATION OF THE BIOMASS DENSITY AND THE AGE STRUCTURE OF THE POPULATIONS, BULL. MATH. BIOPHYS. 30, 553-563.
- DAPSON, R. W. 1971, QUANTITATIVE COMPARISON OF POPULATIONS WITH DIFFERENT AGE STRUCTURES, ANN. ZOOL. FENN. 8(1), 75-79. [PROBABILITY; CONFIDENCE INTERVAL]
- DOI, T. 1949, AN INVESTIGATION ON THE AGE COMPOSITION OF FISH STOCK BY THE ANALYSIS OF REGRESSION. BULL. JAPANESE SOC. SCI. FISH. 15(7), 306-310. [SIMPLE LINEAR REGRESSION]
- EBERHARDT, L. AND R. I. BLOUCH, 1955, ANALYSIS OF PHEASANT AGE RATIOS, TRANS. N. AMER. WILDL. CONF. 20, 357-367.
- EMLEN, J. M. 1970, AGE SPECIFICITY AND ECOLOGICAL THEORY, ECOLOGY 51(4), 588-601.
- FREDRICKSON, A. G. 1971, A MATHEMATICAL THEORY OF AGE STRUCTURE IN SEXUAL POPULATIONS: RANDOM MATING AND MONOGAMOUS MARRIAGE MODELS, MATH. BIOSCI. 10, 117-143.
- GANI, J. 1973, ON THE AGE STRUCTURE OF SOME STOCHASTIC PROCESSES, BULL. INTERN. STAT. INST. 45(1), 434-436. [MULTIPLE BRANCHING PROCESS; LESLIE MATRIX3]
- GOLUBITSKY, M., E. B. KEELER AND M. ROTHSCHILD. 1975, CONVERGENCE OF THE AGE STRUCTURE: APPLICATIONS OF THE PROJECTIVE METRIC, THEOR. POPUL. BIOL. 7(1), 84-93.
- HAYASHI, Y. 1970, STUDIES ON THE MATURITY CONDITION OF THE COMMON SQUID-I. A METHOD OF EXPRESSING MATURITY CONDITION BY NUMERICAL VALUES, BULL. JAPANESE SOC. SCI. FISH. 36(10), 995-999.
- HETT, J. M. 1971, A DYNAMIC ANALYSIS OF AGE IN SUGAR MAPLE SEEDLINGS, ECOLOGY 52(6), 1071-1074. [POWER FUNCTION MODEL]
- KELKER, G. H. 1950, SEX AND AGE CLASS RATIOS AMONG VERTEBRATE POPULATIONS, PROC. UTAH ACAD. SCI., ARTS LETT. 27, 12-21.
- KETCHEN, K. S. 1950, STRATIFIED SUBSAMPLING FOR DETERMINING AGE DISTRIBUTIONS. TRANS. AMER. FISH, SOC. 79, 205-212. [CHI-SQUARE; POLYGON; PROBABILITY3]
- KUTKUHN, J. H. 1963, ESTIMATING ABSOLUTE AGE COMPOSITION OF CALIFORNIA SALMON LANDINGS, CALIF. DEPT. FISH GAME FISH BULL. NO. 120, 42 PP. [TWO-PHASE STRATIFIED SAMPLING; VARIANCE; SAMPLE SIZE]
- LOTKA, A. J. 1922, THE STABILITY OF THE NORMAL AGE DISTRIBUTION, PROC. NATL. ACAD. SCI. U. S. A. 8(11), 339-345.

FIGURE D4. Two Column Format Created by Program TWOCOLM


```

      INTEGER X(4800)
      DATA IFIRST/1/
      IF ( IFIHST ,EQ, 1 ) READ 300,X
      FORMAT(5X,25I3)
      IFIRST = 0
      NUMBER = X ( NUMBER )
      IF ( NUMBER ,LE,0 ) NUMBER=400
      IF ( NUMBER,GT,400) NUMBER=401
      RETURN
    END
X      * * * * * 7 8 9 CARD * * * * *
/ID/BIBLIOGRAPHY
/SELECT/(1-5000)
X00  **END OF RECORD CARD*****
~104254300~4184~3023820228213086082 42 42282 42104104 86254 42282 42184
26242192104130104242300158208300 86282 20158 86 20104 86104104104130254 8 42
51140266 42104278 42140104104104 8158266216104310216202 42 20 8 20130184208
76216329230 42310158 54242 54 54266282104104104 86242158242216184 20 20322322
101310282254192266254254310 42282329266 42 42266 42158322130130 42192 12158140
126310130238266 54329158202104300 42 12254266230282 20202282 54282130130130230
151192 42158 54158266184242184105208216216 54208208208282 20242 12216 76105130
176140192254254192322192322322192192300192158130 20140158192158192158 72105 54
201282208208150 861922661592022022022022422~228228231025113013013026238192
226159192192 20192322216230322 8230192242322193193131159 86230230230 72 86230
251242254105300254254184 86283266131131131131208300 42131105 20193193150 54184
276184105 20105140105184283 20202159283310159159266159159 42 20242159216266 12
301140105131242 54216 54283131 54 42230283254 43105105105105300300105322105106
326193184193310278193 76193322140283 20 20242242242243159 54255300 54310230140
351208310322131131159159283159 12 43 12 54131300266131 76 76216131131131 12159
376 12140283243 86266184 55 86 76 43 76283184159159106106238 86255184 43132 86
401 76300 55315159159283140132106 76283160266216216 21160160184 21216216 43300
426 76 21283193300 21160 21 43202322 43160283 43 5523023027821610630030130130~
451216160184193193 86~4010628331013214013213213221721~1~22~5431043202 21278
476 86 21 21 76 76 21 76 72 72 55 55106217310255 55 87 87106184217185 87160 43
501283 76310 77 87160 21243267106160185140185 21 43 87150150150150 43106140283
526310311255255255 21132132217132132217132284217284 43 55278 12230230301301278
551185267 8278 8267278311 87 43284217 43106255133255267267301243243267301193
576106267106243284267301301255255330106267106 87193243 43 44107 21301160185160
601160322267255107160160284 21 21 21193202193150107 44311160217107107107107107
626245107107185217230230 55 87 87 21140 55 55140160255284140278 21278 55160278
651185278278185140322 77193194322194301322 55133208 72243 72 72208 21278284133
676185311311141 72 22301185161161 87 12 77 55278 55161107255 44217217 72256185
701141301230141230267 55161301256256256256311301231231231208284284161 8185243
726 22161161 44217267231267256243 22311208107161 87161185 22 1231115010716116~
751161256194 55185 22 22 22 56107284284133284150 56 56 72 56107 56311 77141107
776107108108108243 87108133267267 87267243243108243 22 87194161141322202 22194
801208311311209217 44108311202217161311311311 44161202194 6256256256150194133
826108185108133108141141 72311185 56322243301301268361278284108 87202161 56161
851256301 77 56194 77217312218 12161 56 72231218185 22323 56185141 22162 56 44
676133 77243194285268268268256108285 22150244244244244244285 12194 56162185 87
901 22141268209133218231312312256268302108108 77150186 87162 56186108302108108
926302 8108109109109109 44109268244109268268 88268268268109302109268109162
951 56194~0916219419431210910915015088162231162 88 56 44218 56 57150285 44162
976162312 8285231312231244256 12 77 22 13302 13302141 13 13 57 57244141 77329
1001278 88162 22256162162202285323 44 44231109244 13 88162162218162 8323162 13
1026 77 77268 22 72268 13209218162162162163312 22312186 57194133 77238244218323
1051163110 57186141312278209 7721811~30224414131230231231231231231231231244

```

FIGURE D5. Control Cards for Program INDEX with Subroutines CIPHER and XCHANGE and Data Cards for XCHANGE (continued)

```

1076163 88 88163 88 44 88 88 23141 23256312244 44 44203 57313313218256203141 77
1101302 72244285285285285285313110244313218 77 88 44313186163150163285268268302
1126163150163163163163194110244163110110 57 45 45 45 57110 23302 57 23150323231
1151285 45330285285285244186302 8 88 23151302256257257257 45110285323163 57 57110
1176 57209323218141 57 45 45 72 45 72110 23313151 23 23286 23151110110110 72 13
1201110141218110163163 23163218 77286163 57 58111218163 45 23 13 77164164166 8
1226194 58 58 58186 58 58 58209142111244111111194257231269269286231 88 88209209
1251133209209209286 88218244323186164 23151245302164 88286 45 45269269164111186
1276286313133286218133133 78218278164186186 23 58238257257257257111 58 23186219
1301164 88 45 23 23 23 23323 24 24 24 24313313313 24 13 13209209209219269164 88
1326 58 89 45 89286286 78 24142 78219323 24286 45209164111 24164151 24 45209 58
1351142 13 58 58111 58 8 24 24302 78286133231219219219219329219219231219231 58
1376313313 59219 78164164 59 59313164209164278151151257195151151245313313186111
1401111269 59164111 59 59219142 78323219313219313 89151 451113021111111111238 24
1426 24269142279323219302286 78 24245111 59 59323313111 4623821921921923219220 220
1451210 13 24151302112164323323164 24269210195269231 46231232 46142286286232257
1476232 78286151257112112286112112186303112269112269286112112112245112112112112
1501142133133245 13 59 89 59 59323 59 46 59 59220195165232220186 25195 78 25151
1526220245 59 25245303 14314 46314245112220113195245220 25269113323257165113142
1551113165279314165165286 14269269287269257 60 60210210303245151134 25 25134 60
1576287303 73 60165 25 60287210220 25269287232287287287165195186323 25324324 25
1601187187187 25 25220 60165 14 25 25 14303142 8 14 14113220232269287113287203
1626314220113287 60287269245 89165113314287113303245187245151113220165187187165
1651 8 60232303 73287165142314 78220210195287 60165287113113113113187113195287
1676134142245165 14 46279287232134232232232257 73232245245220195270303165 26 26
1701187 26 26 26165 26 60 26 60 60134257220114114257245288288165203203 46134166
1726166166166 78166 26324 26 46 46166270279166134134134134 60166 89 89 89314314
1751166166 60151303134187221314314314 46 78 61 89 46270210 78 78324 78232166 26
1776 26 73 26 26 26314238245166166166166 89114114270 89203187 61166 61270288324
1801324134166 14114166270270270270270114142303 26 26257257258167167 89 61288
1826288288210210221245 89324246303258221303221238288288221258 89258258258258258
1851314314221 14167 89 46 61 61270258258258151 61195114114314314324 89 89 61314
1876288324288167167114167324 26167167187221114303114303114 26134221303288114142
1901314 90288142314288258258288288258324 61114303142258114258115 27 90 61 14 90
1926 27221 61270246 14167288 78289187270115 90329269 90167115115115279203315134
1951195221167203271115115115115115115115167271115115115115 14187303 27 27271246
1976246 8303 90 90115134167135232 78151195195151246315258 90315167167167167167
2001324168 79142 90 61 90 90195221210210210324 61 46187 8259 14168324246324324
2026203168187221195221142 27115 46168271289168 61 79221 61151271 90 46304168143
2051116116116 61 62315 62 90 14246168315289143143210210116116 90 62135289289135
2076135 27 27221143221 79116168116 46304143 90 27 27 73116116116116116116116168116
2101195324315143 62168232168 91168271 91 46168324 27168168135116117117117 62143
2126 79195195195324271 14289289289289 27304304304117117117246 91117271271 8169
2151 46152 62246330210 62143 91 91 91289222222210289289259315315315210222187187
2176152152169169169222289304304117 91169 91289 27117 62117 621171711717 27169 47
2201 62 62289246246117117 27118259246315315315143289203135196246169169222271246
2226290 91169 47118152152246222152152169169 79290 79 27324143 91118118 47259118
2251 27246 15 15247279247259152259196152196196247259271196196304247259247196247
2276169315210187 91 28222 28 28169 28325222169 28169169169 28187325325170152 28
2301170170170170 28188170325 91152152271325 91315 79152196211 91315 47 92259 92
2326143290290152 47135 47203222290196 79 79 28 79 79 79 47 62 47 28203 28 92135
2351 62152 79222 92290118271 92170 8170325170152 9247247247211304170143118170
2376170170315247196325196143315222118316 92316 28 79247247203 28 28 28 79 9 28
2401 28 79 29 79 29 29222325271188325 29222 79 80 62153 29 92 29118 92271247271
2426272272 92211 29170 29 29170316170170143316 62 62259259259259 29 29 29259 9
2451304170290259259 29118 29118116135316316316 80 80 92 80118118316272272272 29
2476 15153290290290290238290290222238232188119304 63279119 15119119171119119119

```

FIGURE D5. Control Cards for Program INDEX with Subroutines CIPHER and XCHANGE and Data Cards for XCHANGE (Continued)

```

2501222232 47 15203 29 29 29 29 15 30 30325 30188 80 30 30 30279135171325 30325
2526171 63188188 63171222290 92 92171 92 47119 47153 80 15119135171 92 92290290
2551188 30238135222 63223223 80211316232232153272171 92119196329 92 93196153171
2576135 30 30 30223325238153247171 63 93153 63119290290272 80 63171171 93 93171
2601153 63304153 47247247247153304304325248223291272 93 93 93119291 30304304 63
2626 63330330171 93 93 30 30 63203 30291119 30 63325203 80171316171316316316316
2651272 63 80171171171119119119119 63196248248248 93 47223223 93291 63 93 93325
2676171 63 30316316232232233305 31272172 73248260260 73 73 73 31 73223 93233233
2701211223223291153291291223223329248248291305305248248305248248153 9 9172172
2726135135135136317 80172317 93305248188188317120120272233223279279305223279291
2751325305291 31 64 15120 93325120172136305 64 31 31 64 31 31 31 31 31 31 94 94
2776 94143 94 94272272 64248 94291172305233291260305120120260143172120144 9172
2801 31260 94153120120172 64196 9172211120317203 15238144 64325 94 94 31144172
282627229127222120291325329 64291 94120233 64291136272136136196196248 31248 31
2851 31223 94326 64 64188196326260120172120120172 9204211223172272291273260260
287630517221121121121121172 94 94279 94 47 15 15 15 15326 15 15329172292 47292
2901 64 64292 95 15 95 95 95 95 95 31 80120120 15173 95305144120292 64 95 16 16
2926305121173173173144 64292 32173211211317 80 80223223211212326224233173292173
2951144 47248 65 32330 95 95136121144144260173 65273153173317196121233233305305
2976305260197197136248121292121144144 95224 95326 32292173 95 65248188 32260273
3001 65 65 16197173329273121121306306306144121 95326173188317260 80326 65 95 65
3026144260136136136224326326173 80 95173 32 81 81 81 81173197 96 47330 96204
3051 65 65 65326326 96 1626030626030630630331326331121273121144 48 48 65 73 65
3076 73326 73 73224292144144 32153153153154248306 9173 32173260197249292197 81
3101197121 65173 65 65 96 9623323323323329273174 48 48317317 48 48 48 48 48 48
3126 65292121174 9273 96 96279249326 96249 32233261273 32 96292317 32 32 66224
3151 3222423823313623422423422422422422422424224 81 81 81 81 81326 32 32 32 32
3176326174326224 81144 48144121 48 96121 33 33 33292 48 16 16121144 96224174 96
3201 33174 48154154197154234121261261261306306 33212188197326224174292261174121
3226122273122 48 48174329174 66174 81273 33174327238306 66 66 81174 81 81 81 81
3251174174 81174249249306174224317327 16 66234273 96174197122306154 33249 66 96
3276261261154261261145261145174175 16 66 9154197212 96136197197197224145175 33
3301175122 66145 97197293 73225261204 82175327175175175 66197175145145145 97175
3326293 49306273122 82 16 16 66 66154273 66175249 33 33 33249 33197212212327 82
3351 33122 49 66 97145306225197306122122327306122327122 97122 97122273122122122
3376327123145145 66279 33 97145197188 97 97188188 33188175273261234175317 97136
3401234261293306293261123175261293307261262273262212 97 16274262262188188225189
3426189 33317 33317317175317317225318327318 66318318318318225204 66154154123327
3451 66 67 33 33123175 82 34 34249 34 34198 97204145189318225234318123123204123
3476123123 34 34 34 34 34249 34 34123123234279189 34 34 34212318225 34318318318
3501 9 67 97274175123175175 16123 67176307 97307293189307123204 34318123 49198
3526198123 97 34 35 35293249262262 35 35204123 82307 16249 67 67 16 97 67 49274
3551 67124307262262124124249124124124262279 9 97 97262 67 67176225 35 35 35 35
3576 35 35 35331 49124293293293234293293293293145124331293293329330331330 98
3601293 73327262198 98176 98225204212198331124330225330 35249225262262262307307
3626307327274274124274274 67176176 9294294294294294294294294294294294294294294
3651225137176 98198198137294249294 49176318 82225 9249294274274 67124145327154
3676154154154154249 82318294294294124225279225189 67319124124124 98327 35 67 98
3701145319 49307307 67 68145 49124 49 82 35 35189 82327 49319 36 98239 16225 68
3726 49 9 9189 82198125274225226137250239234137274 98 73 74176176250327145 36
3751176154 36 74250176176 98327176 98176125 36 66279250 68 68319 98 49319262274
3776176176280145226 16 17295319204250154 36250 98204262226137 98274125176176239
3801189 36295250250125 36 36319319319327176125212 98319319319319189177327 49 36
3826250 49250262 68263263226 49 68 68146 99327250 36177 82125 99177177234307 36
3851 36125 99125125295177177189189 99 68154154155155155155155 49295189 99 99327
3876263155 99 99 49 36239 36 36226 36146189226 68226250177198 74274177319 36125
3901155263226319 36 36307189250 37 99 37 99295295295234295 68280319125295295295

```

FIGURE D5. Control Cards for Program INDEX with Subroutines CIPHER and XCHANGE and Data Cards for XCHANGE (Continued)

```

3926189146295177295295146296125146 99 37125125125125125125125125296126177 74330
3951212212177177204 69204177177 49 50226250250 99177296307307 99 74 99 68263177
3976177 74177319 99198137137198198296274274263263274263263275263275263263263263
4001275307308296126263146126137264308126126155264126126126126126275126137296146
4026308126250296275137137 37 37189 37 68275 57177264126178178 37264 82178146126
4051308178226198178 68178126127137 37126204137234234239275 68 37126127178127280
4976 37 37137 99275146178178319 6h 99 69 69308331331319190 69190127127212296146
4101127328296137 82127100100 37146275198198198127100100100155178 17280 37146 50
4126178306100127100328-7669178280320100178198146127127127226178250251328 37 37
4151 37 3725111'0155146251 38146146146190226212138146296234204296 50204 50100205
4176331205205100205100205 82251178178234328226226280280280280 38 38 38226 38 38
4201251 82 50251 38251251328320275178178226296227178227179155100127227 50 50 69
4226212127179100100179101205205 38 38 38101 69235235101101296320 50101328101101
4251190 38 38328155146147147147127320 17127138296275296328179179 69 63 69328179
4276 17147227127296179101 69275 50212275 69 38179 69179 69 69205 69 69 70328155
4301 74101 38147190190179 38205296179179328297235235179 74297138138 38 50179101
4326128330328 39101280128 50251179508205251190251 39129 39128331205205 50328251
4351326 82 9 9198328 82227275161101 70 70101 39 9198264227128297 39331 50 50
4376297179297179297264264128101227179 50190180180128 17 82213 82 83213328102102
4401 83213 74199199 50308275 70297239102 70227227102102 70138264227227308251 70
4426 39 39 39264 39 83 83 83128180320320180 83 93 83 83138 83 83 83 83 85328297
4451297190180251 70 17 17297199199320320320320297320 50320213213213320213227227
4476320213 83227213213320320 39328 83 74328 39251331180 70227180155147320205 84
4501190328180102275180276 50 51 51 51102329102 70147155297297264264264264 17308
4526147276227180 84180 70331205320128128102 70205 51180 39180180147297128180297
4551 51199235235213227180180297308180 39180297 39102102147128147128147280138156
4576264331331264298180156 51 10329 84181181181213128 39298276276138 51147156 -0
4601 -0 -0 -0 -0 -0 -0 -0 -0 -0 -0 -0 -0 -0 -0 -0 -0 -0 -0 -0 -0 -0 -0 -0
4626 -0 -0 -0 -0 -0 -0 -0 -0 -0 -0 -0 -0 -0 -0 -0 -0 -0 -0 -0 -0 -0 -0 -0
4651 -0 -0 -0 -0 -0 -0 -0 -0 -0 -0 -0 -0 -0 -0 -0 -0 -0 -0 -0 -0 -0 -0 -0
4676 -0 -0 -0 -0 -0 -0 -0 -0 -0 -0 -0 -0 -0 -0 -0 -0 -0 -0 -0 -0 -0 -0 -0
4701 -0 -0 -0 -0 -0 -0 -0 -0 -0 -0 -0 -0 -0 -0 -0 -0 -0 -0 -0 -0 -0 -0 -0
4726 -0 -0 -0 -0 -0 -0 -0 -0 -0 -0 -0 -0 -0 -0 -0 -0 -0 -0 -0 -0 -0 -0 -0
4751 -0 -0 -0 -0 -0 -0 -0 -0 -0 -0 -0 -0 -0 -0 -0 -0 -0 -0 -0 -0 -0 -0 -0
4776 -0 -0 -0 -0 -0 -0 -0 -0 -0 -0 -0 -0 -0 -0 -0 -0 -0 -0 -0 -0 -0 -0 -0
4801322329330 -0 -0 -0 -0 -0 -0 -0 -0 -0 -0 -0 -0 -0 -0 -0 -0 -0 -0 -0 -0 -0 -0
W 0/6/7/8/9 MOHAWK UT-200 END-OF-JOB CARD
W 0/6/7/8/9 MOHAWK UT-200 END-OF-JOB CARD
W 0/6/7/8/9 MOHAWK UT-200 END-OF-JOB CARD

```

*Following the program control card ID, SELECT, data cards are input to INDEX.

FIGURE D5. Control Cards for Program INDEX with Subroutines CIPHER and XCHANGE and Data Cards for XCHANGE (Continued)


```

PROGRAM THOMAS(INPUT,OUTPUT,TAPE5=INPUT,TAPE6=OUTPUT)
DIMENSION A(7,62)
DIMENSION B(7,62)
LOGICAL END
K=332
KK=62
ZZ=1
3  DO 10 J=1,KK
    READ (5,5)(A(I,J),I=1,7)
    IF (EOF(5),NE. 0) END = ,TRUE,
    IF (END) GO TO 30
5  FORMAT (7A5)
10  CONTINUE
    DO 20 J=1,KK
    READ (5,5)(B(I,J),I=1,7)
    IF (EOF(5),NE. 0) END = ,TRUE.
    IF (END) GO TO 30
20  CONTINUE
30  CONTINUE
    K=K+1
    IF (ZZ ,EQ. 0) GO TO 53
50  WRITE (6,60) K
    ZZ=0
    GO TO 55
53  WRITE (6,65) K
    ZZ=1
55  CONTINUE
60  FORMAT(1H1,17X,"INDEX",T89,I3,/)
65  FORMAT(1H1,17X,I3,787,"INDEX",/)
    DO 75 J=1,KK
    WRITE (6,70)(A(I,J),I=1,7),(B(I,J),I=1,7)
70  FORMAT(18X,7A5,3X,7A5)
    IF (END) GO TO 80
75  CONTINUE
    GO TO 3
80  CONTINUE
    STOP
    END

```

FIGURE D6. Program THOMAS to Write Author Index

- AASEN, O., 104,254
 ABAKUKS, A., 300
 ABBOTT, M. B., 184
 ABDELKADER, M. A., 184
 ABERDEEN, J. E. C., 130, $\hat{3}$
 ABOUEL NOUR, A.-R. A., 202
 ABRAHAMSEN, G., 130,282
 ABRAHAM, F. A. H., 300
 ABRAHAMI, G., 86
 ABRAMSON, N. J., 42,104,146,282
 ABROSOV, V. N., 104
 ACOCK, B., 86
 ADAMS, L., 42,254,282
 ADAMS, R. P., 42
 ADISOEMARTO, S., 310
 ADKE, S. R., 184
 ADLER, J., 242
 AHMED, M. S., 192
 AHMED, N. U., 104
 AHUJA, J. C., 130
 AIBA, S., 146
 AIKAWA, H., 104
 AKIN, G. S., 202
 ALBAUGH, D. W., 77
 ALBRECHT, L., 311
 ALCALA, A. C., 242
 ALDERDICE, D. F., 47,158,208,300
 ALDERFER, R. G., 86
 ALDRICH, R. C., 282
 ALEKSEEV, V. V., 20
 ALEXANDER, R. M., 158
 ALIMOV, A. F., 86
 ALLAN, J. D., 20
 ALLEE, W. C., 31
 ALLEN, D. M., 301
 ALLEN, K. R., 8,42,86,104,130,140,
 254,266,278
 ALLEN, R. L., 8
 ALLEN, T. F. H., 158
 ALLEN, W. E., 266
 ALMQUIST, S., 216
 ALPATOV, W. W., 151
 ALSBERG, C. L., 146
 ALTHANN, S. A., 179
 ALVERSON, D. L., 104
 AMADON, D., 310
 AMEN, R. D., 216
 AMIDON, E. L., 202
 AMOS, M. H., 42
 AN DER HEIDEN, U., 220
 ANDERSEN, F. S., 8,20,130,184
 ANDERSEN, K. P., 286
 ANDERSON, A. E., 132
 ANDERSON, A. J. B., 208, $\hat{24}$
 ANDERSON, D. J., 211,216,220, $\hat{11}$
 ANDERSON, D. R., 42,230,329
 ANDERSON, E., 15 $\hat{3}$,310
 ANDERSON, E. R., 97
 ANDERSON, F., 54
 ANDERSON, F. M., 54
 ANDERSON, F. S., 24 $\hat{0}$
 ANDERSON, J. L., 15 $\hat{0}$
 ANDERSON, J. M., 54
 ANDERSON, K. P., 104,266,282
 ANDERSON, L. G., 104
 ANDERSON, M. C., 86,91
 ANDERSON, R. M., 158,242
 ANDERSON, W. N., 39
 ANDERSON, W. W., 28,151
 ANDO, S., 216
 ANDREWARTHA, H. G., 20,72,184,322
 ANDREYEV, V. L., 167,282,310
 ANDRYUSHCHENKO, A. A., 84
 ANDRZEJEWSKI, R., 86,96,192,254,266
 ANGEL, M. V., 310
 ANGELTON, G. M., 42
 ANGULO, J. J., 160
 ANISIMOV, S. I., 98,179
 ANONYMOUS, 42,158,266,282,322,329
 ANSCOMBE, F. J., 42,130
 ANTHONY, E. H., 212
 AOYAMA, I., 12,192
 AR, A., 158,173,175
 ARABINA, I. P., 140
 ARAOZ, J., 310
 ARCHIBALD, E. E. A., 130,238,266
 ARES, J., 54
 ARGENTESI, F., $\hat{29}$
 ARGIERO, L., 15 $\hat{3}$
 ARIMIZU, T., 10 $\hat{4}$,202
 ARMITAGE, P., 300
 ARMSTRONG, J. S., 42
 ARMSTRONG, N. E., 12
 ARNASON, A. N., 81,25 $\hat{0}$,266
 ARNEY, J. D., 230
 ARNOLD, G. W., 231
 ARNOLD, J. C., 220, $\hat{2}$
 ARNOLD, S. J., 20
 ARROYO, G. J., 202
 ARVANITIS, L., 64
 ARVANITIS, L. G., $\hat{1}$,282,292
 ASAMI, T., 297
 ASCE, A. M., 54
 ASHBY, E., 130,224
 ASHBY, W. C., 230
 ASHFORD, J. R., 73,192
 ASKEW, A. J., 42
 ASMUNDSON, V. S., 158
 ATKINSON, L. P., 97
 ATTIWILL, P., 54
 ATTIWILL, P. M., 158
 ATWOOD, E. L., 266
 AUER, C., 72,1 $\hat{3}$ 4,242, $\hat{9}$
 AUERBACH, S. I., 66,90
 AULSTAD, D., 105
 AUSTIN, D. D., 165
 AUSTIN, M. P., 54,20 $\hat{3}$,210,211,216
 AUSTIN, R. W., 168
 AVERY, T. E., 282
 AYALA, F. J., 20,24
 AYERS, J. C., 172,242
 AYZATULLIN, T. A., 12
 BACHI, R., 216
 BACKMANN, R. W., 76

FIGURE D7 Author Index Published in Bibliography

FIGURE D8. Steps Necessary to Produce Text Pages plus Author Index

DISTRIBUTION

No. of
Copies

No. of
Copies

OFFSITE

A. A. Churm 9800 S. Cass Avenue Argonne, IL 60439	Theodore B. Yerke Pacific Southwest Forest and Range Experiment Station U.S. Dept. Agriculture P.O. Box 245 Berkeley, CA 94701
27 ERDA Technical Information Center Richard J. Olson Oak Ridge National Laboratory Environmental Sciences Division Building 2001, Box X Oak Ridge, TN 37830 W. Van Winkle Oak Ridge National Laboratory Environmental Sciences Division Building 2001, Box X Oak Ridge, TN 37830 Rodney Strand Oak Ridge National Laboratory Environmental Sciences Division Building 2001, Box X Oak Ridge, TN 37830 Vincent Schultz Washington State University Zoology Department Pullman, WA 99163 Mauro E. Martignoni Pacific Northwest Forest and Range Experiment Station Forest Science Laboratory U.S. Forest Service Corvallis, OR 97731	Robert M. Russell Pacific Southwest Forest and Range Experiment Station U.S. Dept. Agriculture P.O. Box 245 Berkeley, CA 94701 A. J. Hulbert Lovelace Biomedical and Environmental Research Laboratory Inhalation Toxicology Research Institute P.O. Box 5890 Albuquerque, NM 87115 Michael H. Smith, Director Savannah River Ecology Laboratory Savannah River Plant Drawer E Aiken, SC 29801 J. E. Pinder, III Savannah River Ecology Laboratory Savannah River Plant Drawer E Aiken, SC 29801 Jeff Swinebroad ERDA Division of Biomedical and Environmental Research Washington, D.C. 20545

No. of
Copies

David H. Slade
ERDA Division of Biomedical and
Environmental Research
Washington, D.C. 20545

Robert L. Watters
ERDA Division of Biomedical and
Environmental Research
Washington, D.C. 20545

William O. Forster
ERDA Division of Biomedical
and Environmental Research
Washington, D. C. 20545

John H. Wilson, Jr.
ERDA Division of Biomedical and
Environmental Research
Washington, D.C. 20545

Stan Zellmer
Division of Energy and
Environmental Systems
Argonne National Laboratory
9700 Cass Avenue
Argonne, IL 60439

Ishwar P. Murarka
Division of Environmental
Impact Studies
Argonne National Laboratory
9700 Cass Avenue
Argonne, IL 60439

David N. Edgington
Biological and Environmental
Research Division
Argonne National Laboratory
9700 Cass Avenue
Argonne, IL 60439

Wayne C. Hanson
Los Alamos Scientific Laboratory
H-8 Environmental Studies
P.O. Box 1663
Los Alamos, NM 87545

No. of
Copies

Tom Hakonson
Los Alamos Scientific
Laboratory
H-8 Environmental Studies
P.O. Box 1663
Los Alamos, NM 87545

ONSITE

- 1 ERDA Richland Operations Office
Environmental and Laboratory
Programs Division

Paul F. X. Dunigan, Jr.
Wayne Lei

50 Battelle-Northwest

L. L. Eberhardt
M. L. Cochran (20)
J. M. Thomas (10)
D. H. Fickeisen
J. L. Helbling
R. O. Gilbert
R. H. Sauer
R. Hooper
M. Lee
B. E. Vaughan
C. R. Watson
J. R. Tadlock
J. Wallace
P. J. Dionne
J. C. Fox
Technical Publications (1)
Technical Information Files (3)
Ecosystems Files (3)