NBS TECHNICAL NOTE 1167

U.S. DEPARTMENT OF COMMERCE/National Bureau of Standards

A Computer Data Base System for Indexing Research Papers

NATIONAL BUREAU OF STANDARDS

The National Bureau of Standards[1] was established by an act of Congress on March 3, 1901. The Bureau's overall goal is to strengthen and advance the Nation's science and technology and facilitate their effective application for public benefit. To this end, the Bureau conducts research and provides: (1) a basis for the Nation's physical measurement system, (2) scientific and technological services for industry and government, (3) a technical basis for equity in trade, and (4) technical services to promote public safety. The Bureau's technical work is performed by the National Measurement Laboratory, the National Engineering Laboratory, and the Institute for Computer Sciences and Technology.

THE NATIONAL MEASUREMENT LABORATORY provides the national system of physical and chemical and materials measurement; coordinates the system with measurement systems of other nations and furnishes essential services leading to accurate and uniform physical and chemical measurement throughout the Nation's scientific community, industry, and commerce; conducts materials research leading to improved methods of measurement, standards, and data on the properties of materials needed by industry, commerce, educational institutions, and Government; provides advisory and research services to other Government agencies; develops, produces, and distributes Standard Reference Materials; and provides calibration services. The Laboratory consists of the following centers:

Absolute Physical Quantities[2] — Radiation Research — Chemical Physics — Analytical Chemistry — Materials Science

THE NATIONAL ENGINEERING LABORATORY provides technology and technical services to the public and private sectors to address national needs and to solve national problems; conducts research in engineering and applied science in support of these efforts; builds and maintains competence in the necessary disciplines required to carry out this research and technical service; develops engineering data and measurement capabilities; provides engineering measurement traceability services; develops test methods and proposes engineering standards and code changes; develops and proposes new engineering practices; and develops and improves mechanisms to transfer results of its research to the ultimate user. The Laboratory consists of the following centers:

Applied Mathematics — Electronics and Electrical Engineering[2] — Manufacturing Engineering — Building Technology — Fire Research — Chemical Engineering[2]

THE INSTITUTE FOR COMPUTER SCIENCES AND TECHNOLOGY conducts research and provides scientific and technical services to aid Federal agencies in the selection, acquisition, application, and use of computer technology to improve effectiveness and economy in Government operations in accordance with Public Law 89-306 (40 U.S.C. 759), relevant Executive Orders, and other directives; carries out this mission by managing the Federal Information Processing Standards Program, developing Federal ADP standards guidelines, and managing Federal participation in ADP voluntary standardization activities; provides scientific and technological advisory services and assistance to Federal agencies; and provides the technical foundation for computer-related policies of the Federal Government. The Institute consists of the following centers:

Programming Science and Technology — Computer Systems Engineering.

[1]Headquarters and Laboratories at Gaithersburg, MD, unless otherwise noted; mailing address Washington, DC 20234.
[2]Some divisions within the center are located at Boulder, CO 80303.

A Computer Data Base System for Indexing Research Papers

Judith T. Calabrese
Lawrence J. Kaetzel
Robert A. Glass
George R. Smith

Center for Building Technology
National Engineering Laboratory
National Bureau of Standards
Washington, DC 20234

U.S. DEPARTMENT OF COMMERCE, Malcolm Baldrige, Secretary
NATIONAL BUREAU OF STANDARDS, Ernest Ambler, Director

Issued October 1982

National Bureau of Standards Technical Note 1167
Natl. Bur. Stand. (U.S.), Tech. Note 1167, 102 pages (Oct. 1982)
CODEN: NBTNAE

Supersedes NBS Tech. Note 1123

U.S. GOVERNMENT PRINTING OFFICE
WASHINGTON: 1982

For Sale by the Superintendent of Documents, U.S. Government Printing Office, Washington, DC 20402.
Price $5.50
(Add 25 percent for other than U.S. mailing.)

A COMPUTER DATA BASE SYSTEM FOR INDEXING RESEARCH PAPERS

Judith T. Calabrese, Lawrence J. Kaetzel, Robert A. Glass, and George R. Smith

ABSTRACT

This report represents a significant revision to NBS Technical Note 1123 published in 1980. In that report, the Kaetzel, Glass, Smith (KGS) data base system permitted users to index, edit, classify, and retrieve scientific research paper citations. During the past 15 months, the system was modified and enhanced. All programs are written in standard FORTRAN VII Level I programming language providing transportability among computer systems. Retrieval time has been greatly reduced by changing from a sequential access method to an indexed, directory look-up file structure which allows faster and more efficient random access. The file structure is machine independent. Because of the responsiveness of the extract mode, the one-key retrieval is unnecessary and has been deleted from the revised system. The keyword mode has been replaced by the information mode which provides statistics on authors and keywords. A file maintenance mode has been added to ensure data base integrity. The KGS system has been separated from the larger Publications Data Base and the select data base mode has been removed. Software has been tailored to meet KGS users' needs. Overall, the revised system is faster and uses resources more efficiently than the original data base.

Key words: computer indexing; data base; directory look-up; information retrieval; interactive processing; random access.

TABLE OF CONTENTS

TABLE OF CONTENTS (Continued)

LIST OF FIGURES

PREFACE

The original VGS data base system was developed in conjunction with the Office of Standards Development in the Occupational Safety and Health Administration (OSHA).

Dr. Robert Glass served as NBS leader of the OSHA project to develop new standards for safety and hazard markings. Dr. Glass proposed establishing an automated data base system to organize the documents related to the project. Lawrence Kaetzel and George Smith developed the software to support the original system. Judith Calabrese revised the system design and rewrote the programs to conform to FORTRAN VII standards.

Drs. Jonathan Martin and Dorothy Reed served as reviewers of this paper.

1. INTRODUCTION

NBS Technical Note 1123 published in 1980 described the Kaetzel, Glass, Smith (KGS) data base system. Since the report was published, substantial changes have been made to improve the system. This document describes the revised system.

The most significant modification to the KGS system is a change from a sequential access method to an indexed, directory look-up file structure. This change allows faster and more efficient random access of the master file. All programs have been rewritten to accommodate this enhancement. Because of the responsiveness of the extract mode, the one-key retrieval is unnecessary and has been deleted from the revised system. The keyword mode has been replaced by the information mode which provides statistics on authors as well as keywords. A file maintenance mode has been added to ensure data base integrity. The KGS system has been separated from the larger Publications Data Base and the select data base mode has been removed. Software has been tailored to meet KGS users' needs. All programs are written using standard FORTRAN VII Level I programming language. This change provides greater transportability among computer systems.

The system is designed to help users of medium and small computer systems organize, file, and retrieve documents. A major advantage of implementing a system for personal copies of research papers is that users can now consider all publications in their possession to form a research data base. Furthermore, the system permits the researcher to perform new search strategies on the data base.

This report describes the overall system, its operation, and the computer programs. It allows a user with some familiarity with computers to use or duplicate this system. The software system described enables the user to access the entire data base or subsets of the data base from a computer terminal in a research laboratory or office.

A Perkin-Elmer[1] Model 7/32 minicomputer was used in the development and implementation of the system. Other minicomputer systems can be used to implement the data base system, although minor adjustments in the programs may be necessary.

2. COMPUTER SYSTEM ENVIRONMENT

2.1 HARDWARE

The memory capacity of the Perkin-Elmer computer used is one million bytes. Processing is performed using 32 bit word formats.

---

[1] Reference of computer manufacturer does not constitute an endorsement by NBS.

The computer system hosts a variety of peripheral devices including rotating magnetic disk, magnetic tape, cassette recorders, electrostatic printer/plotter, and a medium speed line printer (see figure 1).

Telecommunications access methods are achieved in one of three ways: 1) commercial telephone network; 2) dedicated short-haul modems; and 3) dedicated cable where short distances (less than 450 feet) are involved.

An asynchronous communications adapter is installed on the computer system and provides a universal hardware communications interface. The ASCII* character data format is used in the transmission of data between the computer system and remote devices.

2.2 SOFTWARE

The system was developed under the "Perkin-Elmer OS/32 MT" software operating system which permits multi-user access to the computer's hardware and software resources.

The data base management software system is entitled the "Kaetzel, Glass, Smith Data Base" (KGS). Software sub-systems used with the KGS data base system are: Perkin-Elmer Multi-Terminal Manager (MTM) and the Perkin-Elmer FORTRAN VII Level I compiler. The function served by MTM is to provide a remote software link between the user's terminal and the production and development facilities of the computer system.

2.3 DATA BASE FILE STRUCTURE

The KGS system uses an indexed, directory look-up file structure (see figure 2). The data base is device independent and consists of a master file, keyword dictionary, author dictionary, keyword index, and author index.

The master file is unsorted and contains specific information for every publication stored. Each record corresponds to a publication. Figure 3 represents the master file record format.

The user is responsible for data representation. However, to ensure integrity and efficient operation, it is important to establish standards for consistent data entry. For example, a recommended format for the author name field is: last name-comma-space-first initial-space-middle initial. The importance of using a consistent format is reflected in the information retrieval mode.

Keyword and author dictionary files provide the mechanism for direct access retrieval of master file records. The keyword dictionary includes each unique keyword entered in the keyword fields of the master file. The dictionary also has pointers to the corresponding master file record (see figure 4). The author dictionary contains the same information for author fields. Both files

---

* American Standard Code for Information Interchange.

are sorted in ascending sequence when the file maintenance mode is run. However, an unsorted overflow area is provided for additional keywords and authors. Dictionary files are recreated using the file maintenance mode when the overflow area is so large that retrieval time is significantly affected.

Index files are used for fast retrieval of dictionary file records. The keyword index contains a sorted list of keyword roots and pointers to the starting dictionary address for a particular range (see figure 5). The author index provides the same information for the author dictionary. For example, the root of keyword "color" is "c". If the first keyword starting with the letter "c" is located at record 17 of the dictionary file, the keyword index would be: c 17. The dictionary is read sequentially, starting with record 17, until the keyword "color" is found. If the keyword is not within the range, the overflow area is read sequentially.

3. MODES OF OPERATION

The KGS data base system is designed to perform a variety of tasks. However, the system can logically be divided into two modes of operation: user mode and file maintenance mode.

3.1 USER MODE

The supervisor program (KGSUPVR) directs the execution of the various processing modules based on input from the user. Refer to figure 6 for software configuration. Program modules are contained within a user software library. Complete listings of the application programs are in appendix C. Upon completion of a run mode, control is returned to the supervisor program. Error conditions during execution of a run mode also result in control returning to the supervisor program.

3.1.1 EDIT Mode (KGSEDIT)

There are four edit modes: APPEND, DELETE, UPDATE, and STOP. All editing is performed by referencing the master file record number. Program CONVERT is used to right-justify and to change the record number to an integer value. Records are numbered from one to the number of the last record on the file. There is no maximum record number except that which is dictated by disk storage capacity.

Dictionary files are simultaneously updated to reflect changes made to the master file. Program KGSPTRUP performs the dictionary pointer update function.

The user is primarily responsible for data integrity. Thus, user review of data upon input is an important aspect of the KGS system. The edit mode displays data elements on the CRT screen for validation by the user. Program KGSDISLN displays the record in a 14 line format. The system routinely checks field size to ensure that an input entry is not too long.

Program KGSADD provides the append function. New records are appended to the master file. Author and keyword fields are automatically added to the dictionary files. In the append mode, each element is requested by first displaying the element name and then a prompt[1] to indicate that input is necessary. The user enters the appropriate information for the entire record. Upon completion, the user validates the record and decides whether or not to save the information.

The nature of data base manipulation dictates that a method be provided for deleting unwanted records from the data base. Program KGSDEL provides this capability. The delete mode marks a record for deletion. The record remains on the data base until the file maintenance procedure is run. Once marked for deletion, however, the record is inaccessible to the user. Author and keyword pointers are deleted from the dictionary files.

KGSUP is used to modify an element or elements of a record which is already on the master file. Changed author or keyword fields are automatically updated on the dictionary files. Data elements are represented in a 14 line format. Each element is referenced by its corresponding line number. The system does not have a character string update capability. Rather, the entire data element is replaced.

The stop command returns program control to the supervisor program.

3.1.2 EXTRACT Mode

Information retrieval and display of the KGS data base is accomplished using the EXTRACT mode (program KGSEXTRT). The user selects from three extract criteria: author, keyword, and all. Information is displayed on the CRT screen and optionally output to a 132-column medium-speed printer using program KGSDSPLY.

Program KGSAUTHX provides the author extract mode. Author index and dictionary files are searched for a match with the input author name. The dictionary file furnishes the locations of all records on the master file containing the requested author name. The master file is accessed directly and the selected records are output in a report format.

Keyword extracts are controlled via program KGSKEYEX. The user enters up to six keywords to be retrieved. Within this mode the user further defines search criteria by requesting a field or string search (see appendix A).

A field search retrieves records from the master file based on keywords entered in the keyword field. The field search also provides a check for UNION and INTERSECTION conditions. If UNION (OR) is selected, the retrieved record must

---

[1] A special character displayed by the computer to indicate that an operator command is expected.

contain at least one of the input keywords. However, if INTERSECTION (AND) is selected, the retrieved record must contain all input keywords. Keyword index and dictionary files are searched for a match with the input keywords and the master file is accessed directly.

The string search (program KGSTRING) sequentially reads the master file searching title and keyword fields for a match with the input string. Program STSRCH compares an input string (maximum 50 characters) with a string up to 100 characters long. When found, the record is output in report format and the search resumes until the entire master file is read.

The ALL mode allows the user to print the entire master file. The output report is sorted by author name. Program KGSALL sequentially reads the author dictionary for direct access record locations on the master file. Duplicate records will be output if a publication has more than one author.

3.1.3 INFORMATION Mode

The information mode provides a sorted summary report of keyword and author fields (see appendix B). The information is taken from the keyword and author dictionaries. The report includes a list of keywords and authors and the number of occurrences for each item.

3.2 FILE MAINTENANCE MODE

The KGS data base is a dynamic structure which must be maintained. Program KGSRENEW in the file maintenance mode removes deleted records from the master file. Programs KGSDCNY and KGSIDX recreate keyword index and dictionary files. Programs KGSDCNYA and KGSIDXA recreate author index and dictionary files. Files are recreated to eliminate overflow fields and to include changes made to the master file. The file maintenance mode is designed to run unattended during non-peak periods.

4. OPERATING PROCEDURES

To activate and operate the KGS system, the user must have a terminal which is compatible with the host computer.

4.1 ACTIVATING THE KGS DATA BASE

Activating the various modes of the KGS data base is accomplished via single commands. The program loading and device assignments are performed using a Perkin-Elmer Command Substitution System (CSS) command file. For a detailed description of CSS refer to the Perkin-Elmer "OS/32 MT Operator's Reference Manual."

The command to activate the user mode is: KGS.
The command to activate file maintenance is: KGSRENEW.

For the user mode, this results in the program being loaded and a message being displayed requesting the desired run mode. The KGSRENEW command loads

and executes all parts of the file maintenance mode. It does not require user input. The command KGSHELP provides basic information on activating the data base.

4.2 USER MODE

When the user mode is activated a message is displayed on the terminal requesting one of four run modes: EDIT, EXTRACT, INFORMATION, or STOP.

4.2.1 EDIT Mode

The user can edit the KGS data base by typing "EDIT". The program asks the user to select append, delete, update, or stop sub-mode. All editing is done through record numbers. The records on the master file are numbered from one to the total number of records on the file.

4.2.1.1 APPEND Sub-Mode

The append mode is used to add new publications to the data base. The user types "APPEND" and the program responds by requesting the author name. The program then prompts for the remainder of the information. When all 14 lines have been entered, the completed record is displayed at the terminal. The user checks the input data and answers the question "SAVE O.K.?". A "YES" answer results in the record being written to the master file. If "NO" is typed, the text is discarded. Dictionary files are automatically updated when a record is saved. The program asks "DO YOU WANT TO CONTINUE IN THE APPEND Mode?" If the user wishes to enter additional publications, "YES" is entered. If the user wishes to return to the EDIT run mode, "NO" is entered.

4.2.1.2 DELETE Sub-Mode

The DELETE mode is used to delete a publication from the master file. The program asks for the number of the record to be deleted. The record is retrieved and displayed at the terminal with the message "DELETE THIS RECORD?". The user responds "YES" or "NO". The program asks "DO YOU WANT TO CONTINUE IN THE DELETE Mode?". The user enters the appropriate response.

4.2.1.3 UPDATE Sub-Mode

The update mode is used to change a record already on the KGS data base. The update program asks for the number of the record to be updated and displays the record on the terminal screen in a 14 line format. The user is asked to "ENTER # OF LINE TO BE UPDATED OR AN * TO END." The user types the appropriate line number and the current contents of the line are displayed. The next line of text typed by the user replaces the old line. The program asks for the next line to be changed.

The user types an asterisk (*) or a carriage return when all changes to the record have been completed. The program displays the entire record and asks "SAVE O.K.?" The user responds "YES" or "NO". The program then asks "DO YOU WISH TO CONTINUE IN UPDATE Mode?". The user answers "YES" or "NO".

4.2.1.4 STOP Sub-Mode

In response to the statement "ENTER EDIT Mode - ", the user types "STOP" or a carriage return to exit from the EDIT mode. Program control returns to the supervisor program.

4.2.2 EXTRACT Mode

The extract mode allows the user to retrieve records from the data base according to specified search criteria. The program asks the user to "ENTER EXTRACT CRITERIA (AUTHOR, KEYWORD, OR ALL)".

4.2.2.1 Author Extract Sub-Mode

Through the author extract sub-mode, the user can retrieve all publications written by a particular author. The program asks the user to "ENTER AUTHORS NAME - (DOE, J. J.) - FORMAT MUST MATCH RECORD". The user types the name of the author in the same format it was entered. The program asks "ALTERNATE PRINT OPTION?". A "YES" response produces a hard-copy report in addition to the terminal display. A "NO" response produces only the terminal output. After the retrieved records are displayed, the program asks "DO YOU WISH TO CONTINUE?". A "NO" response returns control to the supervisor program.

4.2.2.2 Keyword Extract Sub-Mode

The keyword extract sub-mode allows the user to retrieve records containing particular keywords. There are two options available within this mode: field or string search. The program asks: "FIELD OR STRING SEARCH BY KEYWORD?".

The field search produces a report of all publications having the desired keyword in one of the keyword fields. The input keyword must exactly match the keyword field on the master file. On entering the field search, the user is asked to select "UNION (OR) OR INTERSECTION (AND)" search criteria. The program asks the user to "ENTER KEYWORDS". Up to six keywords are entered - one per line. The program asks "ALTERNATE PRINT OPTION?". The user answers "YES" or "NO". The selected publications are displayed on the terminal or concurrently output to a medium-speed printer. When all publications have been displayed, the program asks "DO YOU WISH TO CONTINUE?". A "NO" response returns control to the supervisor program.

The keyword string search allows the user to retrieve publications containing the input string within title or keyword fields. It is a lengthy process because the entire master file is searched. The program asks "ALTERNATE PRINT OPTION?". The user replies "YES" or "NO". The program asks the user to "ENTER KEYWORDS FOR SEARCH". Up to six keywords are entered - one per line. After all selected publications have been displayed or concurrently output to a medium-speed printer, the program asks "DO YOU WISH TO CONTINUE?". A "NO" response returns control to the supervisor program.

4.2.2.3 All Extract Sub-Mode

To produce a listing of the entire master file, the user types "ALL" in response to "ENTER EXTRACT CRITERIA (AUTHOR, KEYWORD, OR ALL)". The listing is sorted by author name and is output to a file (CAL:KGSPRT.DTA) for subsequent printing. After the file has been produced, the program asks "DO YOU WISH TO CONTINUE?". A "NO" response returns control to the supervisor program. (To produce the hard-copy listing, the user types PRINT CAL:KGSPRT.DTA).

4.2.3 INFORMATION Mode

The information mode provides the user with statistics about the KGS data base. The program asks the user to "ENTER INFORMATION DESIRED - AUTHOR, KEY, OR BOTH".

The author information sub-mode prints a sorted list of all authors on the KGS data base and the number of publications on file for that author. The keyword information sub-mode prints a sorted list of all keywords and the number of occurrences of the keyword within the data base. The both option provides separate listings of authors and keywords. After the reports are printed, control returns to the supervisor program.

4.2.4 STOP Mode

The user types "STOP" or a carriage return to exit from the user mode.

4.3 FILE MAINTENANCE MODE

To activate the file maintenance mode, the user types KGSRENEW. No further user input is required. Because of memory requirements, file maintenance mode should be run during non-peak periods.

4.4 ERROR REPORTING

Error conditions during execution of the KGS system fall into two categories: hardware related failures and user errors.

Hardware failures usually occur when a magnetic disk file is accessed. The system reports the condition with an appropriate message and returns control to the supervisor program. A status code indicating the nature of the problem is displayed with the error message. The "Perkin-Elmer FORTRAN VII Reference Manual" contains the list of status codes.

User errors occur when the operator enters an invalid response. An appropriate message is displayed explaining the error. Control remains within the current mode and the user is asked to rekey the response.

4.5 "SIGNOFF" PROCEDURE

Upon completion of KGS data base operations, the operator should sign off the computer system. This is accomplished by entering the command:

SIGNOFF

The computer acknowledges the command by printing the total elapsed time that the user was on-line. To start the system over see Section 4.1.

5. IMPLEMENTATION CONSIDERATIONS

Considerations relating to implementation of the KGS data base software system are described in the following sections. Computer system hardware and software availability will vary among different computing systems. Such factors as memory size and disk storage are critical and require careful consideration before implementation takes place. In situations involving Perkin-Elmer 32 bit computers, the implementation process involves little or no software modifications and less emphasis placed on memory requirements.

5.1 MINIMUM HARDWARE RESOURCES

The amount of memory required to operate the KGS system in the user mode is 75,000 bytes. The file maintenance mode requires 247,750 bytes of memory. Disk storage capacity necessary to operate the KGS system is determined by the number of records in the data base. An approximate calculation to determine disk storage is 900 times the maximum number of records to be stored. This allows for storage of the master file, dictionaries, and indexes.

5.2 SOFTWARE INSTALLATION

Applications programs for the KGS data base will most likely require minor modifications where a computer system other than a Perkin-Elmer 32 bit machine is employed. Changes may be necessary to substitute external or library subroutines.

6. CONCLUSIONS

This report provides a description of an information retrieval system. The advantages of this revised system over the original KGS data base are: better transportability among computer systems, reduced retrieval time, and an enhanced file structure which allows faster and more efficient random access.

Here the user has an upgraded system to aid in developing research projects. The researcher now has increased capability to interrogate computer-stored data bases which provide a file code describing where the document is stored (e.g., file drawer). New search strategies encourage greater efficiency in research efforts.

The revised KGS data base system is compatible with most minicomputer systems. Limitations to implementing the system include a requirement for at least 300,000 bytes of core memory, a FORTRAN VII compiler, and disk capacity sufficient to support the stored files.

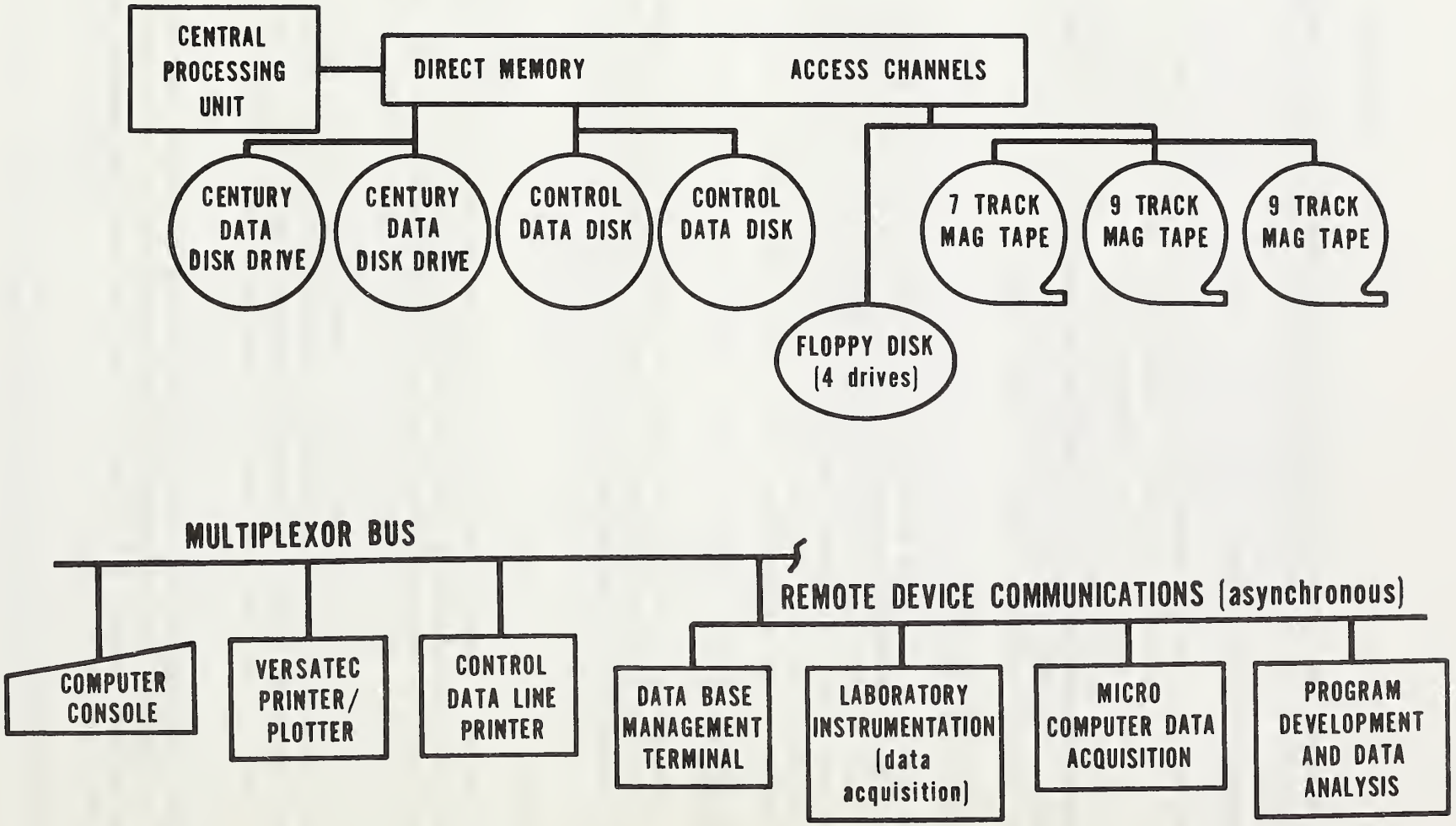


Figure 1. CBT Perkin-Elmer Computer System Configuration

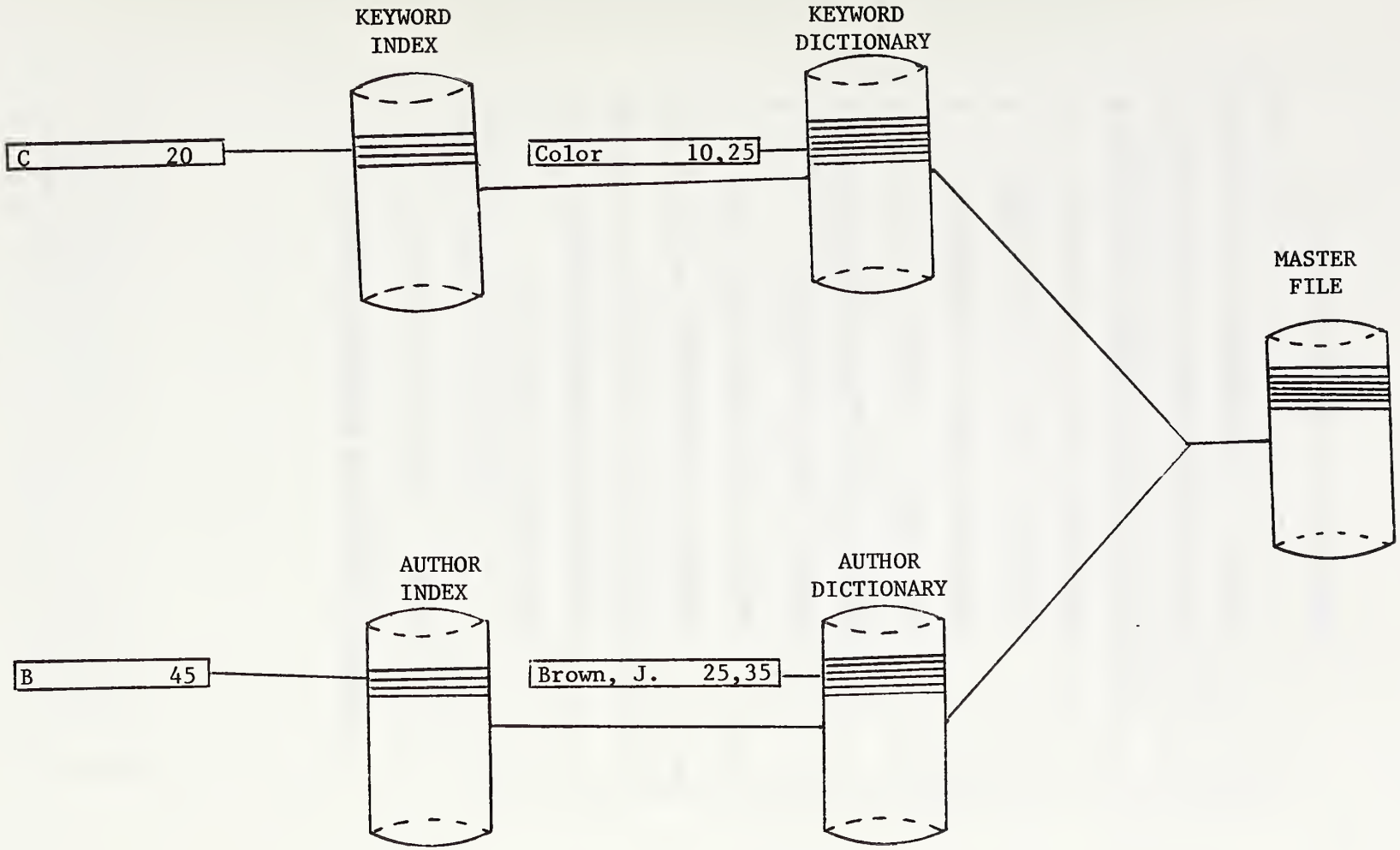
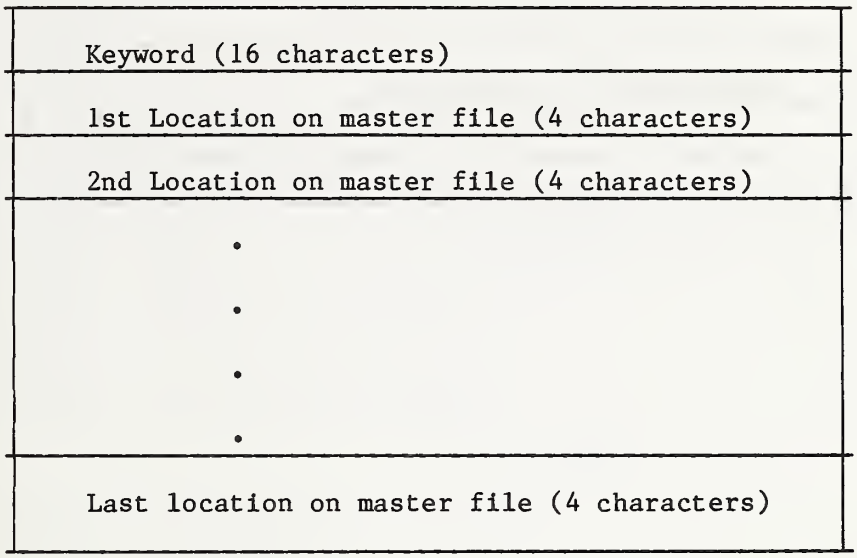


Figure 2. KGS DATA BASE FILE STRUCTURE

Author 1 (32 characters)
Author 2 (32 characters)
Author 3 (32 characters)
Publication Date (12 characters)
Publication Title (100 characters)
Publication Name (32 characters)
Reference (20 characters)
Keyword 1 (16 characters)
Keyword 2 (16 characters)
Keyword 3 (16 characters)
Keyword 4 (16 characters)
Keyword 5 (16 characters)
Keyword 6 (16 characters)
Discipline (16 characters)

Figure 3. KGS Master File Record Format

Keyword Dictionary Record Format



Author Dictionary Record Format

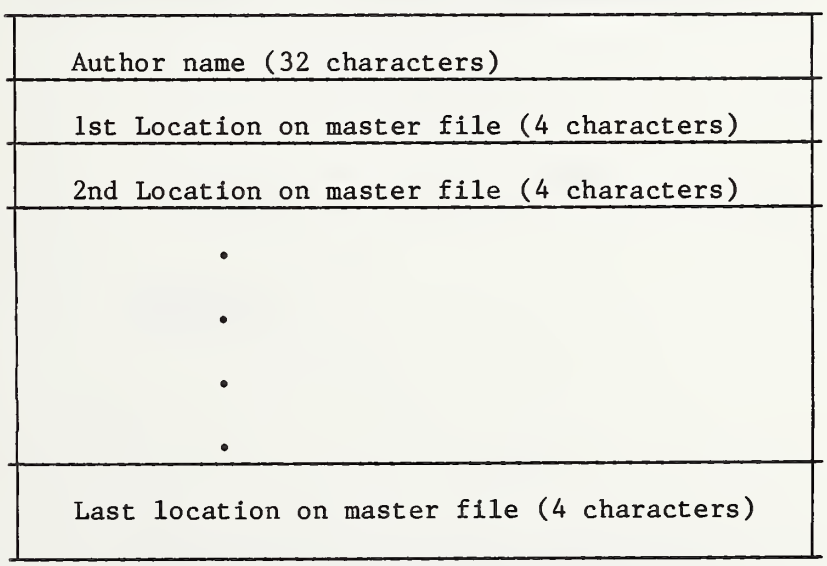
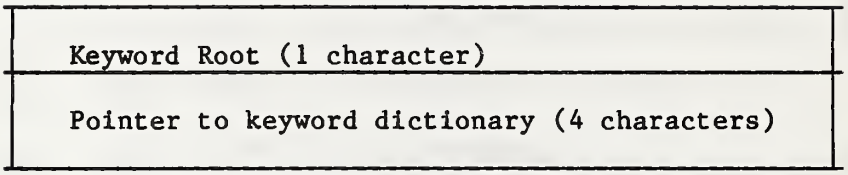


Figure 4. Dictionary Record Formats

Keyword Index Record Format



Author Index Record Format

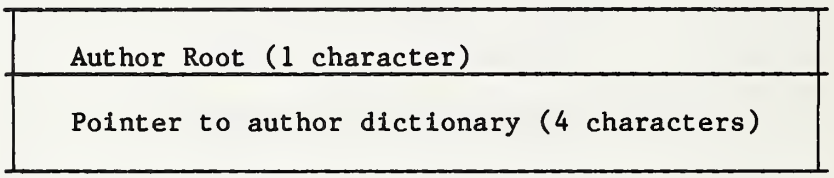


Figure 5. Index Record Formats

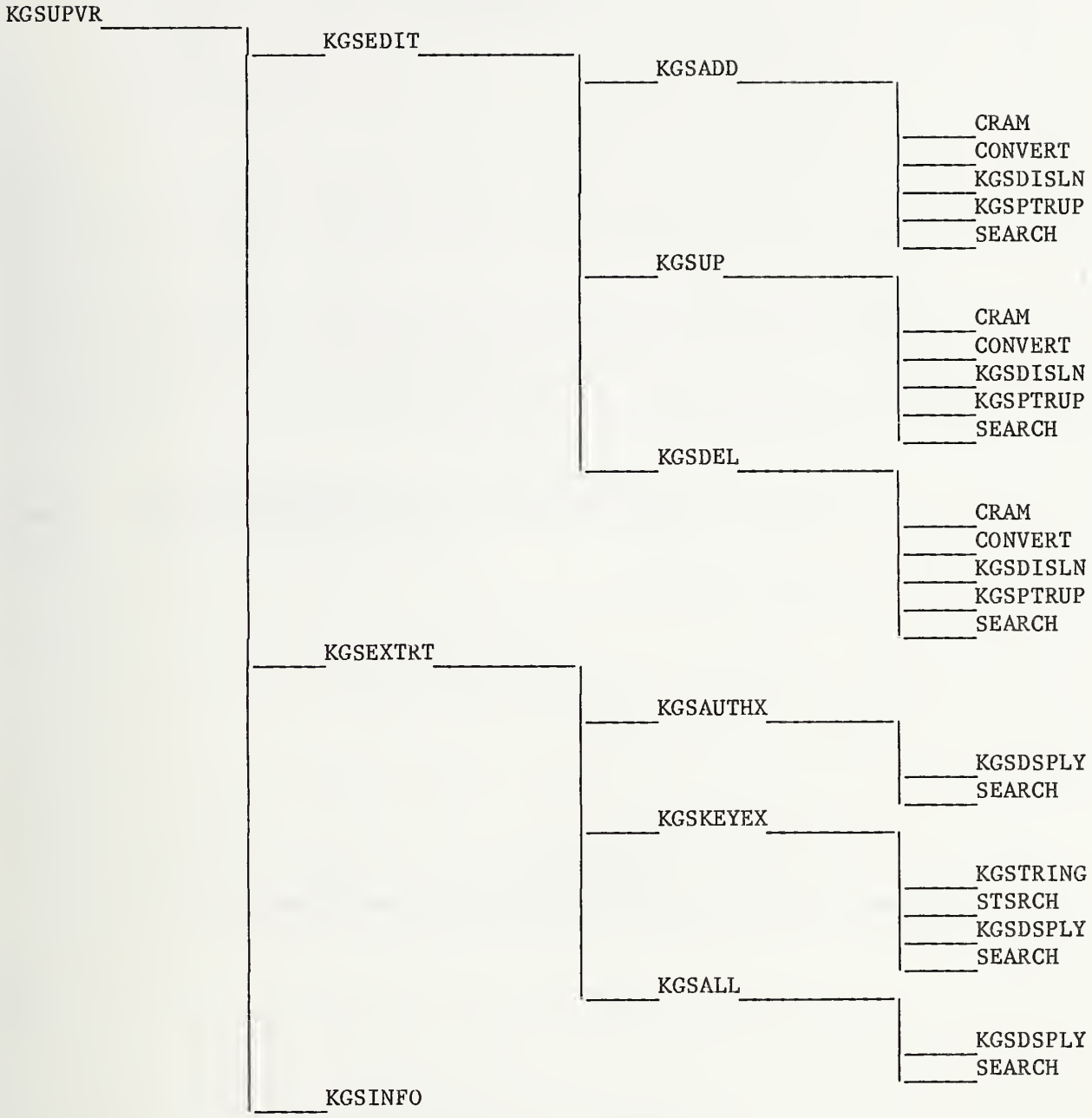


Figure 6. KGS User Mode Program Modules

APPENDIX A

```
KGS


***     FOR ADDITIONAL HELP OR INFORMATION
***     TYPE    KGSHELP
***


        K G S   P U B L I C A T I O N S   D A T A   B A S E
        ***************************************************


                    ENTER DESIRED RUN MODE

        EDIT     EXTRACT     INFORMATION     STOP

EXTRACT
```

K G S P U B L I C A T I O N S

D A T A B A S E R E T R I E V A L M O D E

```
ENTER EXTRACT CRITERIA (AUTHOR, KEY, OR ALL)

KEY

FIELD OR STRING SEARCH BY KEYWORD?
FIELD

UNION (OR) OR INTERSECTION (AND)
OR

ENTER KEYWORDS
ABSOLUTE HUE
XENON FLASHTUBE
COLOUR


KEYWORD NOT IN DICTIONARY - COLOUR

ALTERNATE PRINT OPTION?
NO
- - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - -

KEY WORD/S SELECTED WERE: ABSOLUTE HUE      XENON FLASHTUBE   COLOUR

SEARCH CRITERIA USED: UNION
- - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - -

AUTHOR:              AKITA, M.                              RECORD NO.
AUTHOR(2):           GRAHAM, C. H.
AUTHOR(3):           HSIA, Y.
FILE CODE:           0232
PUBLICATION DATE:    0464
TITLE:               MAINTAINING AN ABSOLUTE HUE IN PRESENCE OF DIFFERE
                     NT BACKGROUND COLORS
PUBLICATION NAME:    VISION RES.
REFERENCE:           4,539-556
- - KEY WORDS - - COLOR PERCEPTION    COLOR CONTRAST      ABSOLUTE HUE
                   BACKGROUND COLOR    LUMINANCE RATIO     BEZOLD-BRUCKE
- - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - -
```

```
- - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - -

KEY WORD/S SELECTED WERE: ABSOLUTE HUE      XENON FLASHTUBE   COLOUR

SEARCH CRITERIA USED: UNION
- - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - -

AUTHOR:              9                                      RECORD NO.   11
AUTHOR(2):           HENDLEY, C. D.
AUTHOR(3):           KULIKOWSKI, J. J.
FILE CODE:
PUBLICATION DATE:    1972
TITLE:               ELECTROPHYSIOLOGICAL AND PSYCHOPHYSICAL RESPONSES
                     TO MODULATION OF CONTRAST OF A GRATING PATTERN
PUBLICATION NAME:    PERCEPTION
REFERENCE:           1,341-349
- - KEY WORDS - - XENON FLASHTUBE     ELECTRONICS         POWER SUPPLIES

- - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - -


DO YOU WISH TO CONTINUE?

NO



                  ENTER DESIRED RUN MODE

         EDIT     EXTRACT     INFORMATION     STOP


STOP
STOP
JUDY     -END OF TASK CODE=   0     CPUTIME=4.442/0.830


***     FOR ADDITIONAL HELP OR INFORMATION
***     TYPE    KGSHELP
***
SIGNOF
ELAPSED TIME=3:44             CPUTIME=4.442/0.830
TIME OFF=12/30/81  11:50:58
```

KGS PUBLICATIONS DATA BASE AUTHOR LIST 12/30/81

AUTHOR	OCCURRENCES	AUTHOR	OCCURRENCES
*	1	BECK, J.	1
9	1	BECK, W. C.	14
ABRAMOV, I.	1	BECK, W. C., GEFFERT, J.	1
ACTKINSON, T. R.	1	BEDELL, H. E.	1
ADAMS, A. J.	1	BEDFORD, R. E.	1
ADAMS, B. B.	1	BEERS, J.	1
AKITA, M.	3	BEGGS, S. S.	1
AKSUGUR, E.	1	BEIJER, L. B.	1
ALBIN, R. E.	1	BELBIN, R. M.	1
ALEXANDER, J. V.	1	BENNETT, C. A.	1
ALLEN, C. J.	1	BERKOVITZ, M.	1
ALLEN, C.J.	1	BERNSTEIN, A.	2
ALLEN, F.	1	BERRY, R. N.	2
ALLEN, M. J.	2	BEXTON, W. H.	1
ALLISON, T.	2	BICK, M. W.	1
ALPERN, M.	4	BIERSNER, R. J.	1
AMBLER, B. A.	1	BINNIE, C. D.	1
AMES, A.	1	BIRREN, F.	1
ANDERSON, D. E.	1	BIRREN, J. E.	1
ANDERSON, E. M. S.	1	BITTERMAN, M. E.	1
ANDREWS, F. M.	1	BLACKWELL, H. R.	7
ANSI	1	BLACKWELL, O. M.	1
ANSTIS, S.	1	BLAKEMORE, C.	1
ANSTIS, S. M.	1	BODINGER, D. M.	1
ARCHEA, J.	1	BODIS-WOLLNER, I.	2
AREND, L. E.	1	BODMANN, H. W.	1
ARMSTRONG, C. E.	2	BONDI, K. R.	1
ARMSTRONG, R.	1	BONEY, D. V.	1
ARTSCHWAGER, E.	2	BOOGAARD, J.	2
BAILEY, R. W.	1	BOOHER, H. R.	1
BAKER, C. A.	1	BOOKER, R. L.	1
BARLOW, H. B.	5	BORING, E. G.	1
BARNES, J. A.	1	BORNSTEIN, M. H.	2
BARTLESON, C. J.	2	BORNSTEIN, W.	1
BARTLETT, F.	1	BORSELLINO, A. I.	1
BARTLETT, N. R.	1	BOUMA, H.	1
BARTZ, A. E.	1	BOUMAN, M. A.	4
BAUER, R. W.	1	BOYCE, P. R.	9
BEARE, A. C.	1	BOYDSTUN, L. E.	1
BEARE, A. N.	1	BOYNTON, R. M.	15

KGS PUBLICATIONS DATA BASE KEYWORD LIST 12/30/81

KEYWORD	OCCURRENCES	KEYWORD	OCCURRENCES
	1	BALLAST DESIGN	1
ABSOLUTE HUE	1	BAR WIDTH	1
ABSORPTION SPECT	1	BEZOLD-BRUCKE	4
ABSTRACTS	2	BIBLIOGRAPHY	4
ACHROMATIC	1	BINOCULAR RIVALR	1
ACHROMATIC CHANN	1	BINOCULAR VISION	2
ACHROMATIC COLOR	1	BIOLOGICAL ASPEC	1
ACOUSTIC DISTORT	1	BIOLOGICAL IMPLI	1
ACOUSTICS	2	BIOMETEROLOGY	1
ACTIFED	1	BLACK BAR	1
ACTINIC EFFECTS	1	BLOCHS LAW	1
ADAPTATION	18	BLUE CONES	1
AGE	7	BLUE/WHITE LIGHT	1
AGE CHANGES	1	BLUR	4
AGE EFFECTS	1	BODY TEMPERATURE	1
AGING	1	BORDER	4
AIR IONS	1	BRIGHT CONSTANCY	1
ALPHA AMYLASE	1	BRIGHT ENHANCE	1
ANIMAL BEHAVIOR	1	BRIGHT GRADIENTS	1
ANIMAL RESEARCH	3	BRIGHT REVERSAL	1
ANOMALOSCOPE	2	BRIGHTNESS	17
ANSI	1	BRIGHTNESS CONTR	1
APERTURE	1	BRIGHTNESS MATCH	1
APPARENT BRIGHTN	3	BRIGHTNESS PERCE	1
APPARENT CONTRAS	2	BUILDINGS	1
APPROACH	1	BUSINESS DISTRIC	1
ARTIFICAL COLOR	1	CALIBRATION	1
ARTIFICIAL IMAGE	1	CATARACT	1
ARTIFICIAL SUNLI	1	CATHODE-RAY TUBE	1
ASTIGMATISM	4	CEREBRAL DOMINAN	1
ASYMMETRY	1	CHLOROLABE	3
ASYNCHRON FLASH	1	CHROMATIC	1
AUDITION	1	CHROMATIC ADAPTA	23
AUDITORY	1	CHROMATIC BORDER	2
AUTONOMIC NERVOU	1	CHROMATIC CHANNE	1
AVERAGE LUMINANC	1	CHROMATIC CONTRA	1
AVIATION	3	CHROMATIC ILLUMI	1
AVIATION SAFETY	1	CHROMATIC INDUCT	1
BACKGROUND COLOR	1	CHROMATIC RESPON	1
BACKWARD MASKING	2	CHROMATIC ROD AC	1

```
C
C
C     PROGRAM KGSUPVR
C
C     WRITTEN BY JUDITH CALABRESE - 19 OCTOBER 1981
C
C     DRIVES THE KGS DATA BASE RETRIEVAL SYSTEM
C
C
C
C             HEADER
C
      WRITE (6,9000,ERR=8000)
C
C             ENTER MODE OF OPERATION
C
50    WRITE (6,9010,ERR=8000)
      READ (6,9020,ERR=8000) IRESP
C
      IF (IRESP.EQ.'EDIT') GO TO 100
      IF (IRESP.EQ.'EXTR') GO TO 200
      IF (IRESP.EQ.'INFO') GO TO 300
      IF (IRESP.EQ.'STOP') GO TO 9999
      IF (IRESP.EQ.'    ') GO TO 9999
      GO TO 8010
C
C             EDIT MODE
C
100   CALL KGSEDIT (ISTAT)
      IF (ISTAT.NE.0) GO TO 9999
      GO TO 50
C
C             EXTRACT MODE
C
200   CALL KGSEXTRT (ISTAT)
      IF (ISTAT.NE.0) GO TO 9999
      GO TO 50
C
C             INFORMATION MODE
C
300   CALL KGSINFO (ISTAT)
      IF (ISTAT.NE.0) GO TO 9999
      GO TO 50
C
C             STOP
C
9999  STOP
C
C
C             ERROR REPORTING AND FORMATTING
C
8000  WRITE (6,8005,ERR=9999)
8005  FORMAT (/1X,'COMMAND DEVICE ERROR')
      GO TO 9999
8010  WRITE (6,8015,ERR=8000)
8015  FORMAT (/1X,'INVALID RESPONSE - PLEASE ENTER AGAIN'/)
```

```
      GO TO 50
C
C
C              I/O FORMATTING
C
9000  FORMAT (//////////10X,'K G S   P U B L I C A T I O N S   ',
     -                      'D A T A    B A S E'//10X,
     -        '**************************************************'//)
9010  FORMAT (/////20X,'ENTER DESIRED RUN MODE'///,
     -        10X,'EDIT     EXTRACT     INFORMATION     STOP'//)
9020  FORMAT (A4)
      END
```

```
C
C
C
      SUBROUTINE KGSEDIT(ISTAT)
C
C     WRITTEN BY JUDITH CALABRESE - 17 SEP 81
C
C     PROVIDES 3 FUNCTIONS FOR THE KGS PUBLICATIONS DATA BASE SYSTEM.
C
C             ADD - ALLOWS THE USER TO ENTER NEW RECORDS.
C
C             DELETE - ALLOWS THE USER TO REMOVE EXISTING RECORDS
C                      FROM THE KGS DATA BASE.
C
C                      NOTE:  THIS FUNCTION ACTUALLY MARKS RECORDS
C                             FOR DELETION RATHER THAN PHYSICALLY
C                             REMOVING THE RECORD.
C
C             UPDATE - ALLOWS THE USER TO UPDATE EXISTING RECORDS.
C
C       IN ALL CASES THE KGS DICTIONARY IS REVISED.
C
C             *****   LOGICAL UNIT ASSIGNMENT   *****
C
C     LU6 - COMMAND DEVICE
C
C             THE FOLLOWING LOGICAL UNITS ARE
C             OPENED AND CLOSED WITHIN SUBROUTINES
C
C     LU1 - KGS DATA BASE
C     LU2 - KGS AUTHOR INDEX
C     LU4 - KGS AUTHOR DICTIONARY
C     LU7 - KGS KEYWORD INDEX
C     LU8 - KGS KEYWORD DICTIONARY
C
C
C
C
C             HEADER
C
      WRITE (6,9020,ERR=8000)
C
C             USER ENTERS MODE OF OPERATION
C
100   WRITE (6,9000,ERR=8000)
      READ (6,9010,ERR=8000) IRESP
      IF (IRESP.EQ.'APPE') GO TO 200
      IF (IRESP.EQ.'DELE') GO TO 300
      IF (IRESP.EQ.'UPDA') GO TO 400
      IF (IRESP.EQ.'STOP') GO TO 9999
      IF (IRESP.EQ.'    ') GO TO 9999
      GO TO 8020
C
C             APPEND
200   CALL KGSADD(ISTAT)
      IF (ISTAT.NE.0) GO TO 9999
```

```
      GO TO 100
C
C             DELETE
C
300   CALL KGSDEL(ISTAT)
      IF (ISTAT.NE.0) GO TO 9999
      GO TO 100
C
C             UPDATE
C
400   CALL KGSUP(ISTAT)
      IF (ISTAT.NE.0) GO TO 9999
      GO TO 100
C
C             RETURN
C
9999  RETURN
C
C             ERROR REPORTING AND FORMATTING
C
8000  WRITE (6,8005,ERR=9999)
8005  FORMAT (/1X,'COMMAND DEVICE ERROR'/)
      GO TO 9999
8020  WRITE (6,8025,ERR=8000)
8025  FORMAT (/1X,'INVALID RESPONSE - PLEASE ENTER AGAIN'/)
      GO TO 100
C
C             I/O FORMATTING
C
9000  FORMAT (/////20X,'ENTER EDIT MODE - '//5X,
     -     'APPEND        DELETE        UPDATE        STOP'//////)
9010  FORMAT (A4)
9020  FORMAT (/////////////////20X,'K G S   P U B L I C A T I O N S',//1X,
     - 10X,'D A T A    B A S E   E D I T   M O D E'////)
      END
```

```
C
C
C
C
C               K G S A D D
C
C
C
C
C
      SUBROUTINE KGSADD(ISTAT)
C     WRITTEN BY JUDITH CALABRESE - 25 SEP 81
C
C     PROVIDES THE APPEND MODE OF THE KGS RETRIEVAL SYSTEM.
C
C     SUBROUTINE CALLED FROM KGSEDIT
C
C              *****   LOGICAL UNIT ASSIGNMENT   *****
C
C     LU1 - KGS AUTHOR INDEX FILE
C     LU2 - KGS AUTHOR DICTIONARY FILE
C     LU5 - KGS DATA BASE
C     LU6 - COMMAND DEVICE
C     LU7 - KGS KEYWORD INDEX FILE
C     LU8 - KGS KEYWORD DICTIONARY FILE
C
C
C
C               KGS DATA BASE DATA ELEMENTS
C
      CHARACTER*32,AUTHOR(3)
      CHARACTER*12,PUBDAT
      CHARACTER*100,TITLE
      CHARACTER*32,PUBNAM
      CHARACTER*20,REFER
      CHARACTER*16,KEYWRD(6)
      CHARACTER*16,DISCPN
C
      CHARACTER*4,BUFFER
      CHARACTER*78,TITLE1
      CHARACTER*35,TITLE2
      CHARACTER*1,AUTIDX(30)
      CHARACTER*1,WRDIDX(30)
      CHARACTER*2,SPACES
      CHARACTER*1,ASTRIX
C
      INTEGER*2 AUTPTR(30),WRDPTR(30),AUTHCT
      INTEGER*2 WRDCT,IREC,KREC
      INTEGER*2 ILINE,ITYPE
C
C
C               INITIALIZE VALUES
C
      SPACES='
      ASTRIX='*'
      AUTHCT=0
      WRDCT=0
```

```
C
C               OPEN KGS INDEX FILES - AUTHOR & KEYWORD
C
      OPEN (1,IOSTAT=ISTAT,ERR=8010,FILE='KGSIDXA.DTA',RECL=5,
     -      SHARE='SRO')
      OPEN (7,IOSTAT=ISTAT,ERR=8010,FILE='KGSIDX.DTA',RECL=5,
     -      SHARE='SRO')
C
C               BUILD INDEX ARRAYS
C               FOR DICTIONARY SEARCH
C
      DO 2 I=1,30
      READ (1,9180,IOSTAT=ISTAT,END=3) AUTIDX(I),AUTPTR(I)
      AUTHCT=AUTHCT+1
2     CONTINUE
3     DO 5 I=1,30
      READ (7,9180,IOSTAT=ISTAT,END=10) WRDIDX(I),WRDPTR(I)
      WRDCT=WRDCT+1
5     CONTINUE
C
C               CLOSE INDEXES
C
10    CLOSE (1)
      CLOSE (7)
C
C
C               OPEN KGS DATA BASE
C
      OPEN (5,IOSTAT=ISTAT,ERR=8010,FILE='KGS',RECL=372,
     -      SHARE='EWO',ACCESS='DIRECT')
C
C               FIND END OF FILE
C
      INQUIRE (5,SIZE=IREC)
C
C               *****   INPUT MODE   *****
C
C               SET INPUT FIELD = SPACES
C
20    DO 30 I=1,3
      AUTHOR(I)='
30    CONTINUE
      PUBDAT='
      TITLE1='
      TITLE2='
      PUBNAM='
      REFER='
      DO 40 I=1,6
      KEYWRD(I)='
40    CONTINUE
      DISCPN='
C
C
C               AUTHORS
C
100   ILINE=1
```

```
      WRITE (6,9000,ERR=8000)
      READ (6,9010,ERR=8000) AUTHOR(1),BUFFER
      IF (AUTHOR(1).EQ.SPACES.OR.AUTHOR(1).EQ.ASTRIX) GO TO 7000
     - IF (BUFFER.NE.SPACES) GO TO 8020
C
120   ILINE=2
      WRITE (6,9020,ERR=8000)
      READ (6,9010,ERR=8000) AUTHOR(2),BUFFER
      IF (AUTHOR(2).EQ.SPACES) GO TO 200
      IF (AUTHOR(2).EQ.ASTRIX) GO TO 7000
      IF (BUFFER.NE.SPACES) GO TO 8020
C
140   ILINE=3
      WRITE (6,9030,ERR=8000)
      READ (6,9010,ERR=8000) AUTHOR(3),BUFFER
      IF (AUTHOR(3).EQ.SPACES) GO TO 200
      IF (AUTHOR(3).EQ.ASTRIX) GO TO 7000
      IF (BUFFER.NE.SPACES) GO TO 8020
C
C             PUBLICATION DATE
C
200   ILINE=4
      WRITE (6,9040,ERR=8000)
      READ (6,9050,ERR=8000) PUBDAT,BUFFER
      IF (PUBDAT.EQ.ASTRIX) GO TO 7000
      IF (BUFFER.NE.SPACES) GO TO 8020
C
C             TITLE
C
      WRITE (6,9060,ERR=8000)
      READ (6,9070,ERR=8000) TITLE1
      IF (TITLE1.EQ.ASTRIX) GO TO 7000
      WRITE (6,9065,ERR=8000)
      READ (6,9075,ERR=8000) TITLE2,BUFFER
      IF (TITLE2.EQ.SPACES) GO TO 350
C
C             REMOVE EXTRA SPACES FROM TITLE FIELD
C
      CALL CRAM(TITLE1,TITLE2)
C
C
C             CREATE 100 CHARACTER TITLE FIELD
C
350   TITLE(1:78)=TITLE1(1:)
      TITLE(79:)=TITLE2(1:)
C
C             PUBLICATION NAME
C
400   ILINE=5
      WRITE (6,9080,ERR=8000)
      READ (6,9010,ERR=8000) PUBNAM,BUFFER
      IF (PUBNAM.EQ.ASTRIX) GO TO 7000
      IF (BUFFER.NE.SPACES) GO TO 8020


 C
              REFERENCE
CC
```

```
500   ILINE=6
      WRITE (6,9090,ERR=8000)
      READ (6,9100,ERR=8000) REFER,BUFFER
      IF (REFER.EQ.ASTRIX) GO TO 7000
      IF (BUFFER.NE.SPACES) GO TO 8020
C
C             KEY WORDS
C
      WRITE (6,9110,ERR=8000)
      DO 650 I=1,6
      READ (6,9120,ERR=8000) KEYWRD(I),BUFFER
      IF (KEYWRD(I).EQ.ASTRIX) GO TO 7000
      IF (KEYWRD(I).EQ.SPACES) GO TO 700
      IF (BUFFER.EQ.SPACES) GO TO 650
      WRITE (6,9140,ERR=8000) KEYWRD(I)
650   CONTINUE
C
C             DISCIPLINE
C
700   ILINE=7
      WRITE (6,9130,ERR=8000)
      READ (6,9120,ERR=8000) DISCPN,BUFFER
      IF (DISCPN.EQ.ASTRIX) GO TO 7000
      IF (BUFFER.NE.SPACES) GO TO 8020
C
C             DISPLAY RECORD
C
      KREC=IREC+1
      CALL KGSDISLN (AUTHOR,PUBDAT,TITLE,PUBNAM,
     -               REFER,KEYWRD,DISCPN,KREC)
C
C
C             SAVE RECORD?
C
C
1000  WRITE (6,9150,ERR=8000)
      READ (6,9160,ERR=8000) IRESP
      IF (IRESP.EQ.'YES ') GO TO 2000
      IF (IRESP.EQ.'NO  ') GO TO 7000
      ILINE=1
      GO TO 8030
C
C             WRITE RECORD TO KGS DATA BASE
C
2000  IREC=IREC+1
      WRITE (5,9170,IOSTAT=ISTAT,ERR=8040,REC=IREC)
     -      (AUTHOR(I),I=1,3),PUBDAT,TITLE,PUBNAM,
     -      REFER,(KEYWRD(J),J=1,6),DISCPN
C
C
C             *****   UPDATE KGS DICTIONARY FILES   *****
C
C
      ITYPE=1
      CALL KGSPTRUP (AUTHOR,KEYWRD,IREC,AUTIDX,WRDIDX,
     -               AUTPTR,WRDPTR,AUTHCT,WRDCT,ITYPE)
```

```
      IF (ITYPE.EQ.9) GO TO 9999
C
C             CONTINUE
C
7000  WRITE (6,9210,ERR=8000)
      READ (6,9160,ERR=8000) IRESP
      IF (IRESP.EQ.'YES ') GO TO 20
      ILINE=2
      IF (IRESP.NE.'NO  ') GO TO 8030
      ISTAT=0
C
C             RETURN
C
9999  CLOSE (5)
      RETURN
C
C
C             ERROR REPORTING AND FORMATTING
C
C
8000  WRITE (6,8005,ERR=9999)
8005  FORMAT (/1X,'COMMAND DEVICE ERROR'/)
      ISTAT=1
      GO TO 9999
8010  WRITE (6,8015,ERR=8000) ISTAT
8015  FORMAT (/1X,'ERROR ON OPENING FILE - IOSTAT = ',I4/)
      GO TO 9999
8020  WRITE (6,8025,ERR=8000)
8025  FORMAT (/1X,'LINE TOO LONG - PLEASE ENTER AGAIN'/)
      GO TO (100,120,140,200,400,500,700),ILINE
8030  WRITE (6,8035,ERR=8000)
8035  FORMAT (/1X,'PLEASE RESPOND YES OR NO'/)
      GO TO (1000,7000),ILINE
8040  WRITE (6,8045,ERR=8000) ISTAT
8045  FORMAT (/1X,'ERROR ON WRITING FILE - IOSTAT = ',I4/)
      GO TO 9999
C
C
C             I/O FORMATTING
C
C
9000  FORMAT (/1X,'ENTER FIRST AUTHOR (DOE, J)'/,1X,
     -        'OR A BLANK LINE TO END'/)
9010  FORMAT (A32,A4)
9020  FORMAT (/1X,'ENTER SECOND AUTHOR IF APPROPRIATE'/)
9030  FORMAT (/1X,'ENTER THIRD AUTHOR IF APPROPRIATE'/)
9040  FORMAT (/1X,'ENTER PUBLICATION DATE'/)
9050  FORMAT (A12,A4)
9060  FORMAT (/1X,'ENTER FIRST LINE OF TITLE')
9065  FORMAT (1X,'ENTER SECOND LINE OF TITLE')
9070  FORMAT (A78)
9075  FORMAT (A35,A4)
9080  FORMAT (/1X,'ENTER PUBLICATION NAME'/)
9090  FORMAT (/1X,'ENTER REFERENCE'/)
9100  FORMAT (A20,A4)
9110  FORMAT (/1X,'ENTER UP TO 6 KEYWORDS - ONE KEYWORD PER LINE'/)
```

```
9120  FORMAT (A16,A4)
9130  FORMAT (/1X,'ENTER DISCIPLINE')
9140  FORMAT (/1X,'TOO LONG!!! - KEYWORD TRUNCATED TO ',A16)
9150  FORMAT (1X,'SAVE O.K.?')
9160  FORMAT (A4)
9170  FORMAT (3A32,A12,A100,A32,A20,6A16,A16)
9180  FORMAT (A1,I4)
9210  FORMAT (1X,'DO YOU WANT TO CONTINUE IN THE APPEND MODE?')
      END
```

```
C
C
C
C
C
C               K G S U P
C
C
C
C
C
C
      SUBROUTINE KGSUP (ISTAT)
C
C     WRITTEN BY JUDITH CALABRESE - 13 OCTOBER 1981
C
C     PROVIDES THE UPDATE MODE OF THE KGS RETRIEVAL SYSTEM.
C
C     SUBROUTINE CALLED FROM KGSEDIT
C
C              *****   LOGICAL UNIT ASSIGNMENTS   *****
C
C     LU1 - KGS AUTHOR INDEX FILE
C     LU2 - KGS AUTHOR DICTIONARY FILE
C     LU5 - KGS DATA BASE
C     LU6 - COMMAND DEVICE
C     LU7 - KGS KEYWORD INDEX FILE
C     LU8 - KGS KEYWORD DICTIONARY FILE
C
C              KGS DATA BASE DATA ELEMENTS
C
      CHARACTER*32,AUTHOR(3)
      CHARACTER*12,PUBDAT
      CHARACTER*100,TITLE
      CHARACTER*32,PUBNAM
      CHARACTER*20,REFER
      CHARACTER*16,KEYWRD(6)
      CHARACTER*16,DISCPN
C
      CHARACTER*32,HAUTHR(3)
      CHARACTER*16,HLDKEY(6)
C
      CHARACTER*32,DAUTHR(3)
      CHARACTER*16,DISKEY(6)
C
      CHARACTER*4,BUFFER
      CHARACTER*78,TITLE1
      CHARACTER*35,TITLE2
C
      CHARACTER*1,AUTIDX(30)
      CHARACTER*1,WRDIDX(30)
C
      CHARACTER*2,SPACES
      CHARACTER*1,ASTRIX
      CHARACTER*4,IRESP
C
      INTEGER*2 AUTPTR(30),WRDPTR(30),AUTHCT,WRDCT
```

```
      INTEGER*2 IREC,ILINE,ITYPE,ENDREC
C
C               INITIALIZE VALUES
C
      SPACES='  '
      ASTRIX='*'
      AUTHCT=0
      WRDCT=0
C

C               OPEN KGS INDEX FILES - AUTHOR & KEYWORD
C
      OPEN (1,IOSTAT=ISTAT,ERR=8010,FILE='KGSIDXA.DTA',RECL=5,
     -      SHARE='SRO')
      OPEN (7,IOSTAT=ISTAT,ERR=8010,FILE='KGSIDX.DTA',RECL=5,
     -      SHARE='SRO')
C
C               BUILD INDEX ARRAYS
C               FOR DICTIONARY SEARCH
C
      DO 2 I=1,30
      READ (1,9000,IOSTAT=ISTAT,END=3) AUTIDX(I),AUTPTR(I)
      AUTHCT=AUTHCT+1
2     CONTINUE
3     DO 5 I=1,30
      READ (7,9000,IOSTAT=ISTAT,END=10) WRDIDX(I),WRDPTR(I)
      WRDCT=WRDCT+1
5     CONTINUE
C
C               CLOSE INDEXES
C
10    CLOSE (1)
      CLOSE (7)
C
C
C               OPEN KGS DATA BASE
C
      OPEN (5,IOSTAT=ISTAT,ERR=8010,FILE='KGS',RECL=372,
     -      SHARE='ERW',ACCESS='DIRECT')
      INQUIRE (5,SIZE=ENDREC)
C
C               *****   UPDATE MODE   *****
C
C
C               USER ENTERS RECORD # TO BE UPDATED
C
20    WRITE (6,9010,ERR=8000)
      READ (6,9020,ERR=8000) IRESP
      IF (IRESP.EQ.ASTRIX) GO TO 9999
      IF (IRESP.EQ.SPACES) GO TO 9999
C
C               CONVERT RECORD # TO INTEGER VALUE
C
      CALL CONVERT (IRESP,IREC)
C
C               CHECK FOR RECORD PAST END-OF-FILE
```

```
      IF (IREC.GT.ENDREC) GO TO 8060
C
C             READ KGS DATA BASE RECORD
C
      READ (5,9030,IOSTAT=ISTAT,ERR=8020,REC=IREC)
     -     (AUTHOR(I),I=1,3),PUBDAT,TITLE,PUBNAM,
     -      REFER,(KEYWRD(J),J=1,6),DISCPN
C
      DO 30 I=1,3
      IF(AUTHOR(I).EQ.'DELEDELEDELEDELEDELEDELEDELEDELE')GO TO 8070
      DAUTHR(I)=AUTHOR(I)
30    CONTINUE
      DO 35 I=1,6
      DISKEY(I)=KEYWRD(I)
35    CONTINUE
C
C
C             INITIALIZE AUTHOR AND KEYWORD FIELDS TO SPACES
C
      DO 70 I=1,3
      HAUTHR(I)=SPACES
70    CONTINUE
      DO 80 I=1,6
      HLDKEY(I)=SPACES
80    CONTINUE
C
C             DISPLAY RECORD
C
40    CALL KGSDISLN(DAUTHR,PUBDAT,TITLE,PUBNAM,
     -              REFER,DISKEY,DISCPN,IREC)
C
C             ENTER LINE # TO BE UPDATED
C
50    WRITE (6,9040,ERR=8000)
      READ (6,9020,ERR=8000) IRESP
      IF (IRESP.EQ.ASTRIX) GO TO 2000
      IF (IRESP.EQ.SPACES) GO TO 2000
      CALL CONVERT (IRESP,ILINE)
C
C             CHECK FOR VALID LINE #
C
      IF (ILINE.LT.1.OR.ILINE.GT.14) GO TO 8030
C
C             UPDATE BASED ON LINE SELECTION
C
      GO TO (100,200,300,400,500,600,700,800,
     -       800,800,800,800,800,1400),ILINE
C
C
C
C             UPDATE FIRST AUTHOR FIELD
C
100   WRITE (6,9050,ERR=8000) AUTHOR(1)
      READ (6,9060,ERR=8000) HAUTHR(1),BUFFER
      IF (BUFFER.NE.SPACES) GO TO 8040
```

```
      DAUTHR(1)=HAUTHR(1)
      GO TO 40
C
C             UPDATE SECOND AUTHOR FIELD
C
200   WRITE (6,9070,ERR=8000) AUTHOR(2)
      READ (6,9060,ERR=8000) HAUTHR(2),BUFFER
      IF (BUFFER.NE.SPACES) GO TO 8040
      DAUTHR(2)=HAUTHR(2)
      GO TO 40
C
C             UPDATE THIRD AUTHOR FIELD
C
300   WRITE (6,9080,ERR=8000) AUTHOR(3)
      READ (6,9060,ERR=8000) HAUTHR(3),BUFFER
      IF (BUFFER.NE.SPACES) GO TO 8040
      DAUTHR(3)=HAUTHR(3)
      GO TO 40
C
C             UPDATE PUBLICATION DATE
C
400   WRITE (6,9090,ERR=8000) PUBDAT
      READ (6,9100,ERR=8000) PUBDAT,BUFFER
      IF (BUFFER.NE.SPACES) GO TO 8040
      GO TO 40
C
C             UPDATE TITLE FIELD
C
500   TITLE1(1:65)=TITLE(1:65)
      TITLE2(1:35)=TITLE(66:100)
C
520   WRITE (6,9110,ERR=8000) TITLE1,TITLE2
      READ (6,9120,ERR=8000) TITLE1
      READ (6,9130,ERR=8000) TITLE2,BUFFER
      IF (TITLE2.EQ.SPACES) GO TO 550
C             REMOVE EXTRA SPACES FROM TITLE FIELD
      CALL CRAM (TITLE1,TITLE2)
C             CREATE 100 CHARACTER TITLE FIELD
550   TITLE(1:78)=TITLE1(1:)
      TITLE(79:)=TITLE2(1:)
      GO TO 40
C
C             UPDATE PUBLICATION NAME
C
600   WRITE (6,9140,ERR=8000) PUBNAM
      READ (6,9060,ERR=8000) PUBNAM,BUFFER
      IF (BUFFER.NE.SPACES) GO TO 8040
      GO TO 40
C
C             UPDATE REFERENCE FIELD
C
700   WRITE (6,9150,ERR=8000) REFER
      READ (6,9160,ERR=8000) REFER,BUFFER
      IF (BUFFER.NE.SPACES) GO TO 8040
      GO TO 40
C
```

```
C               UPDATE KEYWORD FIELDS
C
C
C
800     I=ILINE-7
820     WRITE (6,9170,ERR=8000) KEYWRD(I)
        READ (6,9180,ERR=8000) HLDKEY(I),BUFFER
        DISKEY(I)=HLDKEY(I)
        IF (BUFFER.NE.SPACES) GO TO 8040
        GO TO 40
C
C               UPDATE DISCIPLINE FIELD
C
1400    WRITE (6,9190,ERR=8000) DISCPN
        READ (6,9180,ERR=8000) DISCPN,BUFFER
        IF (BUFFER.NE.SPACES) GO TO 8040
        GO TO 40
C
C
C               SAVE UPDATED RECORD?
C
C
2000    CALL KGSDISLN (DAUTHR,PUBDAT,TITLE,PUBNAM,
     -                  REFER,DISKEY,DISCPN,IREC)
        WRITE (6,9200,ERR=8000)
        READ (6,9020,ERR=8000) IRESP
        IF (IRESP.EQ.'YES ') GO TO 2100
        IF (IRESP.EQ.'NO  ') GO TO 7000
        ILINE=1
        GO TO 8050
C
C
C               UPDATE DICTIONARY FILES
C
C
C               SETUP ARRAYS
C
2100    DO 2140 I=1,3
        IF(HAUTHR(I).EQ.SPACES.AND.DAUTHR(I).NE.SPACES)AUTHOR(I)=SPACES
2140    CONTINUE
C
        DO 2160 I=1,6
        IF(HLDKEY(I).EQ.SPACES.AND.DISKEY(I).NE.SPACES)KEYWRD(I)=SPACES
2160    CONTINUE
C
C               DELETE OLD AUTHORS AND KEYWORDS
C
        ITYPE=2
        CALL KGSPTRUP (AUTHOR,KEYWRD,IREC,AUTIDX,WRDIDX,
     -                  AUTPTR,WRDPTR,AUTHCT,WRDCT,ITYPE)
        IF (ITYPE.EQ.9) GO TO 9999
C
C               ADD NEW AUTHORS TO DICTIONARY
C
        ITYPE=1
```

```
      CALL KGSPTRUP (HAUTHR,HLDKEY,IREC,AUTIDX,WRDIDX,
     -               AUTPTR,WRDPTR,AUTHCT,WRDCT,ITYPE)
      IF (ITYPE.EQ.9) GO TO 9999
C
C
C             REWRITE UPDATED RECORD
C
C
      WRITE (5,9030,IOSTAT=ISTAT,ERR=8020,REC=IREC)
     -      (DAUTHR(I),I=1,3),PUBDAT,TITLE,PUBNAM,
     -       REFER,(DISKEY(J),J=1,6),DISCPN
C
C
C             CONTINUE
C
C
7000  WRITE (6,9210,ERR=8000)
      READ (6,9020,ERR=8000) IRESP
      IF (IRESP.EQ.'YES ') GO TO 20
      ILINE=2
      IF (IRESP.NE.'NO  ') GO TO 8050
      ISTAT=0
C
C             CLOSEOUT
C
9999  CLOSE (5)
      RETURN
C
C
C             ERROR REPORTING AND FORMATTING
C
C
8000  WRITE (6,8005,ERR=9999)
8005  FORMAT (/1X,'COMMAND DEVICE ERROR'/)
      ISTAT=1
      GO TO 9999
8010  WRITE (6,8015,ERR=8000) ISTAT
8015  FORMAT (/1X,'ERROR ON OPENING FILE - IOSTAT = ',I4/)
      GO TO 9999
8020  WRITE (6,8025,ERR=8000) ISTAT
8025  FORMAT (/1X,'ERROR ON READING FILE - IOSTAT = ',I4/)
      GO TO 9999
8030  WRITE (6,8035,ERR=8000)
8035  FORMAT (/1X,'LINE NUMBER IS OUT OF RANGE - '/,
     -         1X,'PLEASE ENTER AGAIN'/)
      GO TO 50
8040  WRITE (6,8045,ERR=8000)
8045  FORMAT (/1X,'LINE TOO LONG!!! - PLEASE ENTER AGAIN'/)
      GO TO (100,200,300,400,520,600,700,820,
     -       820,820,820,820,820,1400),ILINE
8050  WRITE (6,8055,ERR=8000)
8055  FORMAT (/1X,'PLEASE RESPOND YES OR NO'/)
      GO TO (2000,7000),ILINE
8060  WRITE (6,8065,ERR=8000) ENDREC
8065  FORMAT (/1X,'RECORD ENTERED IS PAST END OF FILE'/,
     -         1X,'LAST RECORD # IS ',I4)
```

```
      GO TO 7000
8070  WRITE (6,8075,ERR=8000)
8075  FORMAT (/1X,'RECORD HAS BEEN DELETED'/)
      GO TO 7000
C
C
C              I/O FORMATTING
C
C
9000  FORMAT (A1,I4)
9010  FORMAT (/1X,'ENTER # OF RECORD TO BE UPDATED'/)
9020  FORMAT (A4)
9030  FORMAT (3A32,A12,A100,A32,A20,6A16,A16)
9040  FORMAT (/1X,'ENTER # OF LINE TO BE UPDATED'/,
     -         1X,'OR AN * TO END'/)
9050  FORMAT (/1X,'OLD FIRST AUTHOR IS ',A32/,
     -         1X,'ENTER NEW FIRST AUTHOR'/)
9060  FORMAT (A32,A4)
9070  FORMAT (/1X,'OLD SECOND AUTHOR IS ',A32/,
     -         1X,'ENTER NEW SECOND AUTHOR'/)
9080  FORMAT (/1X,'OLD THIRD AUTHOR IS ',A32/,
     -         1X,'ENTER NEW THIRD AUTHOR'/)
9090  FORMAT (/1X,'OLD PUBLICATION DATE IS ',A12/,
     -         1X,'ENTER NEW PUBLICATION DATE'/)
9100  FORMAT (A12,A4)
9110  FORMAT (/1X,'OLD TITLE IS ',/1X,A78/,1X,A35/,
     -         1X,'ENTER NEW TITLE'/)
9120  FORMAT (A78)
9130  FORMAT (A35,A4)
9140  FORMAT (/1X,'OLD PUBLICATION NAME IS ',A32/,
     -         1X,'ENTER NEW PUBLICATION NAME'/)
9150  FORMAT (/1X,'OLD REFERENCE IS ',A20/,
     -         1X,'ENTER NEW REFERENCE'/)
9160  FORMAT (A20,A4)
9170  FORMAT (/1X,'OLD KEYWORD IS ',A16/,1X,'ENTER NEW KEYWORD'/)
9180  FORMAT (A16,A4)
9190  FORMAT(/1X,'OLD DISCIPLINE IS ',A16/,1X,'ENTER NEW DISCIPLINE'/)
9200  FORMAT (/1X,'SAVE O.K.?'/)
9210  FORMAT (/1X,'DO YOU WANT TO CONTINUE IN UPDATE MODE?'/)
      END
```

```
C
C
C
C
C
C
C               C R A M
C
C
C
C
C
C
      SUBROUTINE CRAM (TITLE1,TITLE2)
C
C     WRITTEN BY JUDITH CALABRESE - 25 SEPTEMBER 1981
C
C     THIS SUBROUTINE REMOVES EXTRA SPACES FROM FIRST LINE OF THE
C     TITLE FIELD.
C     IT ALSO JOINS THE SECOND LINE TO THE FIRST.
C
C
C               ARGUMENTS
C
C     TITLE1 - FIRST LINE OF TITLE FIELD (78 CHARACTERS).
C     TITLE2 - SECOND LINE OF TITLE (35 CHARACTERS).
C
C
      CHARACTER*78,TITLE1
      CHARACTER*35,TITLE2
C
      CHARACTER*2,SPACES
      CHARACTER*2,HOLD
C
C
C               INITIALIZE VALUES
C
      SPACES='  '
      IBYTE=0
C
C               REMOVE SPACES FROM FIRST LINE
C
      DO 100 I=1,77
      IBYTE=IBYTE+1
      LAST=IBYTE+1
      HOLD(1:2)=TITLE1(IBYTE:LAST)
      IF (HOLD.EQ.SPACES) GO TO 50
      ICTR=0
      GO TO 100
50    J=IBYTE+1
      TITLE1(IBYTE:)=TITLE1(J:)
      IBYTE=IBYTE-1
      LAST=LAST-1
      ICTR=ICTR+1
100   CONTINUE
C
C               CALCULATE # OF TRAILING SPACES ON FIRST LINE
```

```
      IFILL=78-(ICTR-1)
      IF (IFILL.GT.78) GO TO 300
      IFILL2=IFILL+1
      IBYTE=0
C
C             ADD SECOND LINE TO FIRST
C
      DO 200 I=1,35
      IBYTE=IBYTE+1
      HOLD(1:1)=TITLE2(IBYTE:IBYTE)
      TITLE1(IFILL:IFILL)=HOLD(1:1)
      J=IBYTE+1
      TITLE2(IBYTE:)=TITLE2(J:)
      IFILL=IFILL+1
      IFILL2=IFILL2+1
      IF (IFILL.GT.78) GO TO 300
      IBYTE=IBYTE-1
C
200   CONTINUE
C
C
C          CHECK FOR LINE TOO LONG
C
C
300   HOLD(1:2)=TITLE2(23:24)
      IF (HOLD.NE.SPACES) GO TO 8010
C
C             RETURN
C
9999  RETURN
C
C
C            ERROR REPORTING & FORMATTING
C
C
8000  WRITE (6,8005,ERR=9999)
8005  FORMAT (1X,'COMMAND DEVICE ERROR')
      GO TO 9999
8010  WRITE (6,8015,ERR=8000) TITLE1,TITLE2
8015  FORMAT (/1X,'TITLE TOO LONG - TRUNCATED TO'/,1X,A78/,1X,A22/)
      GO TO 9999
      END
```

```
C
C
C
C
C
C
C                K G S D E L
C
C
C
C
C
      SUBROUTINE KGSDEL(ISTAT)
C
C     WRITTEN BY JUDITH CALABRESE - 9 OCTOBER 1981
C
C     ALLOWS THE USER TO DELETE RECORDS FROM
C     THE KGS DATA BASE
C
C     NOTE:  RECORDS ARE MARKED FOR DELETION.
C            THIS SUBROUTINE DOES NOT RESEQUENCE THE DATA BASE.
C
C     THIS SUBROUTINE IS CALLED FROM KGSEDIT
C
C              *****   LOGICAL UNIT ASSIGNMENTS   *****
C
C     LU1 - KGS AUTHOR INDEX FILE
C     LU2 - KGS AUTHOR DICTIONARY FILE
C     LU5 - KGS DATA BASE
C     LU6 - COMMAND DEVICE
C     LU7 - KGS KEYWORD INDEX FILE
C     LU8 - KGS KEYWORD DICTIONARY FILE
C
C
C              KGS DATA BASE DATA ELEMENTS
C
      CHARACTER*32,AUTHOR(3)
      CHARACTER*12,PUBDAT
      CHARACTER*100,TITLE
      CHARACTER*32,PUBNAM
      CHARACTER*20,REFER
      CHARACTER*16,KEYWRD(6)
      CHARACTER*16,DISCPN
C
      CHARACTER*1,AUTIDX(30)
      CHARACTER*1,WRDIDX(30)
C
      CHARACTER*4,IRESP
C
      INTEGER*2 AUTPTR(30),WRDPTR(30),AUTHCT,WRDCT
      INTEGER*2 IREC,ITYPE,ENDREC
C
C              INITIALIZE VALUES
C
      AUTHCT=0
      WRDCT=0
C
```

```
C             OPEN KGS INDEX FILES - AUTHOR & KEYWORD
C
      OPEN (1,IOSTAT=ISTAT,ERR=8010,FILE='KGSIDXA.DTA',RECL=5,
     -      SHARE='SRO')
      OPEN (7,IOSTAT=ISTAT,ERR=8010,FILE='KGSIDX.DTA',RECL=5,
     -      SHARE='SRO')
C
C             BUILD INDEX ARRAYS
C             FOR DICTIONARY SEARCH
C
      DO 2 I=1,30
      READ (1,9000,IOSTAT=ISTAT,END=3) AUTIDX(I),AUTPTR(I)
      AUTHCT=AUTHCT+1
2     CONTINUE
3     DO 5 I=1,30
      READ (7,9000,IOSTAT=ISTAT,END=10) WRDIDX(I),WRDPTR(I)
      WRDCT=WRDCT+1
5     CONTINUE
C
C             CLOSE INDEXES
C
10    CLOSE (1)
      CLOSE (7)
C
C
C             OPEN KGS DATA BASE
C
      OPEN (5,IOSTAT=ISTAT,ERR=8010,FILE='KGS',RECL=372,
     -      SHARE='ERW',ACCESS='DIRECT')
      INQUIRE (5,SIZE=ENDREC)
C
C
C             *****   DELETE MODE   *****
C
C
C             ACCEPT RECORD NUMBER FROM USER
C
20    WRITE (6,9010,ERR=8000)
      READ (6,9020,ERR=8000) IRESP
C
C             CONVERT RECORD NUMBER TO INTEGER VALUE
C
      CALL CONVERT (IRESP,IREC)
C
C             CHECK FOR RECORD PAST END-OF-FILE
C
      IF (IREC.GT.ENDREC) GO TO 8050
C
C             READ KGS DATA BASE RECORD
C
      READ (5,9030,IOSTAT=ISTAT,ERR=8020,REC=IREC)
     -     (AUTHOR(I),I=1,3),PUBDAT,TITLE,PUBNAM,
     -     REFER,(KEYWRD(J),J=1,6),DISCPN
      IF(AUTHOR(1).EQ.'DELEDELEDELEDELEDELEDELEDELEDELE')GO TO 8040
C
C             DISPLAY RECORD
```

```
C
      CALL KGSDISLN (AUTHOR,PUBDAT,TITLE,PUBNAM,
     -               REFER,KEYWRD,DISCPN,IREC)
C
C             DELETE THIS RECORD?
C
      WRITE (6,9040,ERR=8000)
      READ (6,9020,ERR=8000) IRESP
      IF (IRESP.NE.'YES ') GO TO 7000
C
C             UPDATE KGS DICTIONARY FILES
C
      ITYPE=2
      CALL KGSPTRUP (AUTHOR,KEYWRD,IREC,AUTIDX,WRDIDX,
     -               AUTPTR,WRDPTR,AUTHCT,WRDCT,ITYPE)
      IF (ITYPE.EQ.9) GO TO 9999
C
C             REWRITE DELETED RECORD
C
      DO 100 I=1,3
      AUTHOR(I)='DELEDELEDELEDELEDELEDELEDELEDELE'
100   CONTINUE
      WRITE (5,9030,IOSTAT=ISTAT,ERR=8030,REC=IREC)
     -      (AUTHOR(I),I=1,3),PUBDAT,TITLE,PUBNAM,
     -       REFER,(KEYWRD(J),J=1,6),DISCPN
C
C             CONTINUE
C
7000  WRITE (6,9050,ERR=8000)
      READ (6,9020,ERR=8000) IRESP
      IF (IRESP.EQ.'YES ') GO TO 20
      IF (IRESP.NE.'NO  ') GO TO 8060
      ISTAT=0
C
C
C             RETURN
C
C
9999  CLOSE (5)
      RETURN
C
C
C             ERROR REPORTING AND FORMATTING
C
C
8000  WRITE (6,8005,ERR=9999)
8005  FORMAT (/1X,'COMMAND DEVICE ERROR'/)
      ISTAT=1
      GO TO 9999
8010  WRITE (6,8015,ERR=8000) ISTAT
8015  FORMAT (/1X,'ERROR ON OPENING FILE - IOSTAT = ',I4/)
      GO TO 9999
8020  WRITE (6,8025,ERR=8000) ISTAT
8025  FORMAT (/1X,'ERROR ON READING FILE - IOSTAT = ',I4/)
      GO TO 9999
8030  WRITE (6,8035,ERR=8000) ISTAT
```

```
8035  FORMAT (/1X,'ERROR ON WRITING FILE - IOSTAT = ',I4/)
      GO TO 9999
8040  WRITE (6,8045,ERR=8000)
8045  FORMAT (/1X,'RECORD HAS ALREADY BEEN DELETED'/)
      GO TO 7000
8050  WRITE (6,8055,ERR=8000) ENDREC
8055  FORMAT (/1X,'RECORD ENTERED IS PAST END-OF-FILE'/,
     -         1X,'LAST RECORD NUMBER IS ',I4/)
      GO TO 7000
8060  WRITE (6,8065,ERR=8000)
8065  FORMAT (/1X,'PLEASE RESPOND YES OR NO'/)
      GO TO 7000
C
C
C             I/O FORMATTING
C
C
9000  FORMAT (A1,I4)
9010  FORMAT (/1X,'ENTER # OF RECORD TO BE DELETED'/)
9020  FORMAT (A4)
9030  FORMAT (3A32,A12,A100,A32,A20,6A16,A16)
9040  FORMAT (/1X,'DELETE THIS RECORD?'/)
9050  FORMAT (/1X,'DO YOU WANT TO CONTINUE IN DELETE MODE?'/)
      END
```

```
C
C
C
C
C
C
C               C O N V E R T
C
C
C
C
C
      SUBROUTINE CONVERT (ASCII,INTGER)
C
C     WRITTEN BY JUDITH CALABRESE - 13 OCTOBER 1981
C
C     CONVERTS ASCII VALUE TO INTEGER
C     AFTER RIGHT JUSTIFYING FIELD
C
      CHARACTER*4,ASCII
      CHARACTER*4,FIELD
      CHARACTER*1,HLDCHR
C
      INTEGER*2 INTGER
C
      FIELD=ASCII
      J=1
C
C               RIGHT JUSTIFY ASCII FIELD
C
      DO 20 I=4,1,-1
      J=J+1
      HLDCHR(1:1)=ASCII(I:I)
      IF (HLDCHR.NE.' ') GO TO 40
      FIELD='    '
      FIELD(J:)=ASCII(1:)
20    CONTINUE
C
C               CONVERT FIELD TO INTEGER
C
40    DECODE (FIELD,50) INTGER
50    FORMAT (I4)
C
C               RETURN
C
      RETURN
      END
```

```
C
C
C
C
C
C
C
C
C
C          K G S D I S L N
C
C
C
C
C
      SUBROUTINE KGSDISLN (AUTHOR,PUBDAT,TITLE,PUBNAM,
     -                REFER,KEYWRD,DISCPN,IREC)
C
C     WRITTEN BY JUDITH CALABRESE - 25 SEPTEMBER 1981
C
C     DISPLAYS KGS RECORD WITH LINE # FOR
C             ADD
C             DELETE
C             UPDATE
C     MODES.
C
C
      CHARACTER*32,AUTHOR(3)
      CHARACTER*12,PUBDAT
      CHARACTER*100,TITLE
      CHARACTER*32,PUBNAM
      CHARACTER*20,REFER
      CHARACTER*16,KEYWRD(6)
      CHARACTER*16,DISCPN
C
      CHARACTER*50,TITLE1(2)
C
      INTEGER*2 IREC
C
C
C             SPLIT TITLE FIELD
C
      TITLE1(1)(1:50)=TITLE(1:50)
      TITLE1(2)(1:50)=TITLE(51:)
C
C
C
      WRITE (6,9000,ERR=8000) IREC
      WRITE (6,9010,ERR=8000) AUTHOR(1)
      WRITE (6,9020,ERR=8000) AUTHOR(2)
      WRITE (6,9030,ERR=8000) AUTHOR(3)
      WRITE (6,9035,ERR=8000) PUBDAT
      WRITE (6,9040,ERR=8000) (TITLE1(I),I=1,2)
      WRITE (6,9050,ERR=8000) PUBNAM
      WRITE (6,9060,ERR=8000) REFER
      WRITE (6,9070,ERR=8000)
```

```
      WRITE (6,9080,ERR=8000) KEYWRD(1)
      WRITE (6,9090,ERR=8000) KEYWRD(2)
      WRITE (6,9100,ERR=8000) KEYWRD(3)
      WRITE (6,9110,ERR=8000) KEYWRD(4)
      WRITE (6,9120,ERR=8000) KEYWRD(5)
      WRITE (6,9130,ERR=8000) KEYWRD(6)
      WRITE (6,9140,ERR=8000) DISCPN
C
C             RETURN
C
9999  RETURN
C
C
C             ERROR REPORTING AND FORMATTING
C
C
8000  WRITE (6,8005,ERR=9999)
8005  FORMAT (/1X,'COMMAND DEVICE ERROR'/)
      GO TO 9999
C
C
C             I/O FORMATTING
C
C
9000  FORMAT (/1X,'**********   RECORD # ',I4,'  **********')
9010  FORMAT (1X,'LINE  1 - ',A32)
9020  FORMAT (1X,'LINE  2 - ',A32)
9030  FORMAT (1X,'LINE  3 - ',A32)
9035  FORMAT (1X,'LINE  4 - ',A12)
9040  FORMAT (1X,'LINE  5 - ',A50/,
     -       11X,A50)
9050  FORMAT (1X,'LINE  6 - ',A32)
9060  FORMAT (1X,'LINE  7 - ',A20)
9070  FORMAT (1X,'*****   KEYWORDS   *****')
9080  FORMAT (1X,'LINE  8 - ',A16)
9090  FORMAT (1X,'LINE  9 - ',A16)
9100  FORMAT (1X,'LINE 10 - ',A16)
9110  FORMAT (1X,'LINE 11 - ',A16)
9120  FORMAT (1X,'LINE 12 - ',A16)
9130  FORMAT (1X,'LINE 13 - ',A16)
9140  FORMAT (/1X,'LINE 14 - ',A16/)
C
C
      END
```

```
C
C
C
C
C
C             K G S P T R U P
C
C
C
C
C
      SUBROUTINE KGSPTRUP(AUTHOR,KEYWRD,IREC,AUTIDX,WRDIDX,
     -                    AUTPTR,WRDPTR,AUTHCT,WRDCT,ITYPE)
C
C     WRITTEN BY JUDITH CALABRESE - 7 OCTOBER 1981
C
C
C     UPDATES POINTERS ON KGS DICTIONARY FILES
C
C             *****   LOGICAL UNIT ASSIGNMENTS   *****
C
C     LU2 - KGS AUTHOR DICTIONARY FILE
C     LU8 - KGS KEYWORD DICTIONARY FILE
C     LU6 - COMMAND DEVICE
C
C             ARGUMENTS PASSED
C
C     AUTHOR - READ FROM KGS DATA BASE
C     KEYWRD - READ FROM KGS DATA BASE
C     IREC   - RECORD NUMBER ON KGS DATA BASE
C     AUTIDX - AUTHOR INDEX (POINTS TO DICTIONARY FILE)
C     WRDIDX - KEYWORD INDEX (POINTS TO DICTIONARY FILE)
C     AUTHCT - COUNT OF CELLS ON AUTHOR INDEX
C     WRDCT  - COUNT OF CELLS ON KEYWORD INDEX
C     ITYPE  - 1 = APPEND MODE
C              2 = DELETE MODE
C              9 = RETURN CODE ERROR
C
C
C
      CHARACTER*32,AUTHOR(3)
      CHARACTER*16,KEYWRD(6)
C
      CHARACTER*1,AUTIDX(30)
      CHARACTER*1,WRDIDX(30)
C
      CHARACTER*1,HLDCHR
      CHARACTER*2,SPACES
C
      CHARACTER*32,AUTHDC
      CHARACTER*16,WRDDCN
C
      INTEGER*2 AUTPTR(30),WRDPTR(30),ICELL,RECNO(60)
      INTEGER*2 KREC,WRDCT,AUTHCT,IREC,IOVFLW,ITYPE
C
C             INITIALIZE VALUES
C
```

```
      SPACES='  '
C
C
C             *****   AUTHOR   *****
C
C
C             OPEN AUTHOR DICTIONARY
C
      OPEN (2,IOSTAT=ISTAT,ERR=8010,FILE='KGSAUTH.DTA',RECL=232,
     -      SHARE='ERW',ACCESS='DIRECT')
C
C             FIND START OF OVERFLOW RECORDS IN DICTIONARY
C
      READ (2,9000,IOSTAT=ISTAT,REC=1) IOVFLW
C             CHECK FOR END-OF-FILE CONDITION
      IF (ISTAT.EQ.31.OR.ISTAT.EQ.32) THEN
        IOVFLW=1
        WRITE (2,9000,IOSTAT=ISTAT,ERR=8040,REC=1) IOVFLW
      END IF
C
C             SEARCH INDEX FOR POINTER TO DICTIONARY
C
      DO 700 I=1,3
      IF (AUTHOR(I).EQ.SPACES) GO TO 700
      HLDCHR(1:)=AUTHOR(I)(1:1)
      CALL SEARCH (AUTIDX,AUTHCT,HLDCHR,ICELL)
      IF (ICELL.EQ.0) GO TO 200
      KREC=AUTPTR(ICELL)
C
C             FIND AUTHOR IN DICTIONARY
C
100   READ (2,9010,IOSTAT=ISTAT,REC=KREC) AUTHDC,(RECNO(J),J=1,50)
C             CHECK FOR END-OF-FILE CONDITION
      IF (ISTAT.EQ.31.OR.ISTAT.EQ.32) GO TO 300
      IF (ISTAT.NE.0) GO TO 8020
      KREC=KREC+1
      IF (AUTHDC.LT.AUTHOR(I)) GO TO 100
      IF (AUTHDC.EQ.AUTHOR(I).AND.ITYPE.EQ.1) GO TO 400
      IF (AUTHDC.EQ.AUTHOR(I).AND.ITYPE.EQ.2) GO TO 500
C
C             SEARCH OVERFLOW FOR MATCH
C
200   KREC=IOVFLW+1
220   READ (2,9010,IOSTAT=ISTAT,REC=KREC) AUTHDC,(RECNO(J),J=1,50)
C             CHECK FOR END-OF-FILE CONDITION
      IF (ISTAT.EQ.31.OR.ISTAT.EQ.32) GO TO 300
      IF (ISTAT.NE.0) GO TO 8020
      KREC=KREC+1
      IF (AUTHDC.NE.AUTHOR(I)) GO TO 220
      IF (AUTHDC.EQ.AUTHOR(I).AND.ITYPE.EQ.1) GO TO 400
      IF (AUTHDC.EQ.AUTHOR(I).AND.ITYPE.EQ.2) GO TO 500
C
C             AUTHOR NOT IN DICTIONARY
C
300   RECNO(1)=IREC
      DO 350 J=2,50
```

```
      RECNO(J)=0
350   CONTINUE
      KREC=KREC+1
      GO TO 600
C
C             ADD RECORD NUMBER TO AUTHOR IN DICTIONARY
C
400   DO 450 J=1,50
      IF (RECNO(J).NE.0) GO TO 450
      RECNO(J)=IREC
      GO TO 600
450   CONTINUE
      ICELL=I
      GO TO 8030
C
C             DELETE RECORD NUMBER FROM DICTIONARY
C
500   DO 550 J=1,50
      IF (RECNO(J).EQ.0) GO TO 600
      IF (RECNO(J).NE.IREC) GO TO 550
      DO 520 K=J,50
      RECNO(K)=RECNO(K+1)
      IF (RECNO(K).EQ.0) GO TO 600
520   CONTINUE
550   CONTINUE
C
C             REWRITE AUTHOR DICTIONARY RECORD
C
600   KREC=KREC-1
      WRITE(2,9010,IOSTAT=ISTAT,ERR=8040,REC=KREC)
     -      AUTHOR(I),(RECNO(J),J=1,50)
C
700   CONTINUE
C
C             *****   KEYWRD   *****
C
C
C             OPEN KEYWRD DICTIONARY
C
      OPEN (8,IOSTAT=ISTAT,ERR=8010,FILE='KGSKEY.DTA',RECL=256,
     -      SHARE='ERW',ACCESS='DIRECT')
C
C             FIND START OF OVERFLOW RECORDS IN DICTIONARY
C
      READ (8,9000,IOSTAT=ISTAT,REC=1) IOVFLW
C             CHECK FOR END-OF-FILE CONDITION
      IF (ISTAT.EQ.31.OR.ISTAT.EQ.32) THEN
        IOVFLW=1
        WRITE (8,9000,IOSTAT=ISTAT,ERR=8040,REC=1) IOVFLW
      END IF
C
C             SEARCH INDEX FOR POINTER TO DICTIONARY
C
      DO 1700 I=1,6
      IF (KEYWRD(I).EQ.SPACES) GO TO 1700
      HLDCHR(1:)=KEYWRD(I)(1:1)
```

```
      CALL SEARCH (WRDIDX,WRDCT,HLDCHR,ICELL)
      IF (ICELL.EQ.0) GO TO 1200
      KREC=WRDPTR(ICELL)
C
C             FIND KEYWRD IN DICTIONARY
C
1100  READ (8,9020,IOSTAT=ISTAT,REC=KREC) WRDDCN,(RECNO(J),J=1,60)
C             CHECK FOR END-OF-FILE CONDITION
      IF (ISTAT.EQ.31.OR.ISTAT.EQ.32) GO TO 1300
      IF (ISTAT.NE.0) GO TO 8020
      KREC=KREC+1
      IF (WRDDCN.LT.KEYWRD(I)) GO TO 1100
      IF (WRDDCN.EQ.KEYWRD(I).AND.ITYPE.EQ.1) GO TO 1400
      IF (WRDDCN.EQ.KEYWRD(I).AND.ITYPE.EQ.2) GO TO 1500
C
C             SEARCH OVERFLOW FOR MATCH
C
1200  KREC=IOVFLW+1
1220  READ (8,9020,IOSTAT=ISTAT,REC=KREC) WRDDCN,(RECNO(J),J=1,60)
C             CHECK FOR END-OF-FILE CONDITION
      IF (ISTAT.EQ.31.OR.ISTAT.EQ.32) GO TO 1300
      IF (ISTAT.NE.0) GO TO 8020
      KREC=KREC+1
      IF (WRDDCN.NE.KEYWRD(I)) GO TO 1220
      IF (WRDDCN.EQ.KEYWRD(I).AND.ITYPE.EQ.1) GO TO 1400
      IF (WRDDCN.EQ.KEYWRD(I).AND.ITYPE.EQ.2) GO TO 1500
C
C             KEYWRD NOT IN DICTIONARY
C
1300  RECNO(1)=IREC
      DO 1350 J=2,60
      RECNO(J)=0
1350  CONTINUE
      KREC=KREC+1
      GO TO 1600
C
C             ADD RECORD NUMBER TO KEYWRD IN DICTIONARY
C
1400  DO 1450 J=1,60
      IF (RECNO(J).NE.0) GO TO 1450
      RECNO(J)=IREC
      GO TO 1600
1450  CONTINUE
      ICELL=I
      GO TO 8030
C
C             DELETE RECORD NUMBER FROM DICTIONARY
C
1500  DO 1560 J=1,60
      IF (RECNO(J).EQ.0) GO TO 1600
      IF (RECNO(J).NE.IREC) GO TO 1560
      DO 1520 K=J,60
      RECNO(K)=RECNO(K+1)
      IF (RECNO(K).EQ.0) GO TO 1600
1520  CONTINUE
1560  CONTINUE
```

```
C
C             REWRITE KEYWRD DICTIONARY RECORD
C
1600  KREC=KREC-1
      WRITE(8,9020,IOSTAT=ISTAT,ERR=8040,REC=KREC)
     -      KEYWRD(I),(RECNO(J),J=1,60)
C
1700  CONTINUE
C
C
C             RETURN
C
C
9999  CLOSE (2)
      CLOSE (8)
      RETURN
C
C             ERROR REPORTING AND FORMATTING
C
8000  WRITE (6,8005,ERR=9999)
8005  FORMAT (/1X,'COMMAND DEVICE ERROR'/)
      ITYPE=9
      GO TO 9999
8010  WRITE (6,8015,ERR=8000) ISTAT
8015  FORMAT (/1X,'ERROR ON OPENING FILE - IOSTAT = ',I4/)
      ITYPE=9
      GO TO 9999
8020  WRITE (6,8025,ERR=8000) ISTAT
8025  FORMAT (/1X,'ERROR ON READING FILE - IOSTAT = ',I4/)
      ITYPE=9
      GO TO 9999
8030  WRITE (6,8035,ERR=8000) AUTHOR(ICELL)
8035  FORMAT (/1X,'DICTIONARY OVERFLOW ON AUTHOR - ',A32/)
      ITYPE=9
      GO TO 9999
8040  WRITE (6,8045,ERR=8000) ISTAT
8045  FORMAT (/1X,'ERROR ON WRITING FILE - IOSTAT = ',I4/)
      ITYPE=9
      GO TO 9999
C
C             I/O FORMATTING
C
9000  FORMAT (I4)
9010  FORMAT (A32,50I4)
9020  FORMAT (A16,60I4)
      END
```

```
C
      SUBROUTINE KGSEXTRT(ISTAT)
C
C     WRITTEN BY JUDITH CALABRESE - 21 JULY 1981
C
C     MAIN PROGRAM OF THE KGS EXTRACT MODE.
C
C     PROVIDES FOR RETRIEVAL OF KGS DATA BASE RECORDS
C     BASED ON THE FOLLOWING PARAMETERS:
C
C             AUTHOR
C             KEYWORDS (1 TO 6)
C
C
C     LOGICAL UNIT ASSIGNMENTS
C
C     LU6-COMMAND DEVICE
C
C
C             WRITE MAIN HEADING
C
      WRITE (6,9000,ERR=8000)
C
C             ACCEPT PARAMETERS FROM USER
C
100   WRITE (6,9010,ERR=8000)
      READ (6,9020,ERR=8000) IPARAM
      IF (IPARAM.EQ.'AUTH') GO TO 200
      IF (IPARAM.EQ.'KEY ') GO TO 300
      IF (IPARAM.EQ.'ALL ') GO TO 400
      IF (IPARAM.EQ.'    ') GO TO 9999
      GO TO 8010
C
C             AUTHOR EXTRACT
C
200   CALL KGSAUTHX(ISTAT)
      IF (ISTAT.NE.0) GO TO 9999
      GO TO 500
C
C             KEY EXTRACT
C
300   CALL KGSKEYEX(ISTAT)
      IF (ISTAT.NE.0) GO TO 9999
      GO TO 500
C
C             ALL EXTRACT
C
400   CALL KGSALL(ISTAT)
      IF (ISTAT.NE.0) GO TO 9999
      GO TO 500
C
C             CONTINUE
C
500   WRITE (6,9030,ERR=8000)
      READ (6,9020,ERR=8000) IRESP
      IF (IRESP.EQ.'YES ') GO TO 100
```

```
      IF (IRESP.EQ.'NO  ') GO TO 9999
      GO TO 8020
C
C             CLOSEOUT
C
9999  RETURN
C
C             ERROR REPORTING AND FORMATTING
C
8000  WRITE (6,8005,ERR=9999)
8005  FORMAT (/1X,'COMMAND DEVICE ERROR'/)
      GO TO 9999
8010  WRITE (6,8015,ERR=9999)
8015  FORMAT (/1X,'INVALID RESPONSE - PLEASE ENTER AGAIN'/)
      GO TO 100
8020  WRITE (6,8025,ERR=8000)
8025  FORMAT (/1X,'PLEASE RESPOND YES OR NO'/)
      GO TO 500
C
             I/O FORMATTING
C
9000  FORMAT (/////////////20X,'K G S   P U B L I C A T I O N S',//1X,
     - 10X,'D A T A   B A S E   R E T R I E V A L   M O D E'////)
9010  FORMAT (/1X,'ENTER EXTRACT CRITERIA (AUTHOR, KEY, OR ALL)'/)
9020  FORMAT (A4)
9030  FORMAT (/1X,'DO YOU WISH TO CONTINUE?'/)
      END
```

```
C
C
C
C
C                    K G S A U T H X
C
C
C
      SUBROUTINE KGSAUTHX(ISTAT)
C
C     WRITTEN BY JUDITH CALABRESE - 31 AUGUST 1981
C
C     SUBROUTINE CALLED BY KGSEXTRT.
C
C     SEARCHES THE AUTHOR FIELDS FOR A MATCH
C     WITH A USER-SUPPLIED AUTHOR NAME.
C
C             * LOGICAL UNIT ASSIGNMENTS *
C
C     LU1-KGSINDEX AUTHOR DATA FILE
C     LU2-KGSAUTH.DTA (KGS AUTHOR DICTIONARY FILE)
C     LU3-PRINT DEVICE
C     LU4-KGS DATA BASE
C     LU6-COMMAND DEVICE
C
      CHARACTER*32,INAUTH
      CHARACTER*32,DCNNAM
      CHARACTER*16,INKEY(6)
      CHARACTER*1,INDEX(30)
      CHARACTER*1,HLDCHR
C
C             DATA ELEMENTS READ FROM KGS DATA BASE RECORD
C
      CHARACTER*32,AUTHOR(3)
      CHARACTER*12,PUBDAT
      CHARACTER*100,TITLE
      CHARACTER*32,PUBNAM
      CHARACTER*20,REFER
      CHARACTER*16,KEYWRD(6)
      CHARACTER*16,DISCPN
C
      INTEGER IDATE(3)
      INTEGER*2 RECNO(50),POINTR(30),ISTAT,ILINE
      INTEGER*2 ICELL,IOVFLW,IPRTSW,IPRTCT,IDXCT
      INTEGER*2 IOVFL1,IPAGE,KREC
C
C             INITIALIZE VALUES
C
      IDXCT=0
      ISUB=0
      IPRTSW=0
      IPRTCT=0
      IPAGE=1
      IANDOR=0
C
C             DATE ROUTINE
```

```
      CALL DATE (IDATE)
C
C             ACCEPT AUTHOR NAME FROM USER
C
      WRITE (6,9000,ERR=8000)
      READ (6,9010,ERR=8000) INAUTH
      IF (INAUTH.EQ.'                                    ') GO TO 9999
      HLDCHR=INAUTH
C
C             BUILD INDEX ARRAY FROM INDEX FILE
C
      OPEN (1,IOSTAT=ISTAT,ERR=8010,FILE='KGSIDXA.DTA',RECL=5,
     -      SHARE='SRO')
      DO 100 I=1,30
      READ (1,9020,IOSTAT=ISTAT,ERR=8020,END=150) INDEX(I),POINTR(I)
      IDXCT=IDXCT+1
100   CONTINUE
150   CLOSE (1,IOSTAT=ISTAT,ERR=8030)
C
C             OPEN DICTIONARY
C
      OPEN (2,IOSTAT=ISTAT,ERR=8010,FILE='KGSAUTH.DTA',RECL=232,
     -     SHARE='SRO',ACCESS='DIRECT')
C
C             FIND START OF OVERFLOW RECORDS ON DICTIONARY
C
      READ (2,9030,IOSTAT=ISTAT,REC=1) IOVFLW
C             CHECK FOR END-OF-FILE CONDITION
      IF (ISTAT.EQ.31.OR.ISTAT.EQ.32) GO TO 8060
      IF (ISTAT.NE.0) GO TO 8020
C
C             BINARY SEARCH OF INDEX
C
      CALL SEARCH (INDEX,IDXCT,HLDCHR,ICELL)
      IREC=POINTR(ICELL)
      IF (ICELL.GT.0) GO TO 200
      GO TO 220
C
C             FIND AUTHOR IN DICTIONARY
C
200   READ (2,9040,IOSTAT=ISTAT,REC=IREC)
     -     DCNNAM,(RECNO(I),I=1,50)
C             CHECK FOR END-OF-FILE CONDITION
      IF (ISTAT.EQ.31.OR.ISTAT.EQ.32) GO TO 8040
      IF (ISTAT.NE.0) GO TO 8020
      IREC=IREC+1
      IF (DCNNAM.LT.INAUTH) GO TO 200
      IF (DCNNAM.EQ.INAUTH) GO TO 300
C
C             SEARCH DICTIONARY OVERFLOW FOR AUTHOR
C
220   IOVFL1=IOVFLW
250   READ (2,9040,IOSTAT=ISTAT,REC=IOVFL1)
     -     DCNNAM,(RECNO(I),I=1,50)
C             CHECK FOR END-OF-FILE CONDITION
```

```
      IF (ISTAT.EQ.31.OR.ISTAT.EQ.32) GO TO 8040
      IF (ISTAT.NE.0) GO TO 8020
      IOVFL1=IOVFL1+1
      IF (DCNNAM.EQ.INAUTH) GO TO 300
      GO TO 250
C
C             CLOSE AUTHOR DICTIONARY
C             OPEN KGS DATA BASE
C
300   CLOSE (2,IOSTAT=ISTAT,ERR=8030)
      OPEN (5,IOSTAT=ISTAT,ERR=8010,FILE='KGS',RECL=372,SHARE='SRO',
     -      ACCESS='DIRECT')
C
C
C             ALTERNATE PRINT OPTION
C
C             HARD COPY OUTPUT - IPRTSW=1
C             OUTPUT ONLY ON TERMINAL - IPRTSW=0
C
350   WRITE (6,9050,ERR=8000)
      READ (6,9060,ERR=8000) IRESP
      IF (IRESP.EQ.'NO  ') GO TO 400
      IF (IRESP.NE.'YES ') GO TO 8050
      OPEN (3,IOSTAT=ISTAT,ERR=8010,FILE='PR2:')
      IPRTSW=1
C
C             RETRIEVE RECORDS FROM KGS DATA BASE
C
400   DO 450 I=1,50
      IF (RECNO(I).EQ.0) GO TO 500
      READ(5,9070,IOSTAT=ISTAT,ERR=8020,REC=RECNO(I))(AUTHOR(J),J=1,3),
     - PUBDAT,TITLE,PUBNAM,REFER,(KEYWRD(K),K=1,6),DISCPN
C
C             DISPLAY RECORD
C
      KREC=RECNO(I)
      CALL KGSDSPLY(INKEY,INAUTH,IDATE,KREC,
     -     IANDOR,IPRTSW,IPAGE,IPRTCT,
     - AUTHOR,PUBDAT,TITLE,PUBNAM,REFER,KEYWRD,DISCPN)
450   CONTINUE
C
C             CLOSEOUT
C
500   CLOSE (5,IOSTAT=ISTAT,ERR=8030)
      CLOSE (3,IOSTAT=ISTAT,ERR=8030)
      ISTAT=0
C
C             RETURN
C
9999  RETURN
C
C             ERROR REPORTING AND FORMATTING
C
8000  WRITE (6,8005,ERR=9999)
8005  FORMAT (/1X,'COMMAND DEVICE ERROR'/)
      GO TO 9999
```

```
8010  WRITE (6,8015,ERR=8000) ISTAT
8015  FORMAT (/1X,'ERROR ON OPENING FILE - IOSTAT = ',I4/)
      PAUSE
      GO TO 9999
8020  WRITE (6,8025,ERR=8000) ISTAT
8025  FORMAT (/1X,'ERROR ON READING FILE - IOSTAT = ',I4/)
      PAUSE
      GO TO 9999
8030  WRITE (6,8035,ERR=8000) ISTAT
8035  FORMAT (/1X,'ERROR ON CLOSING FILE - IOSTAT = ',I4/)
      PAUSE
      GO TO 9999
8040  WRITE (6,8045,ERR=8000) INAUTH
8045  FORMAT (/1X,'AUTHOR NAME NOT IN DICTIONARY - ',A32/)
      GO TO 500
8050  WRITE (6,8055,ERR=8000)
8055  FORMAT (/1X,'PLEASE RESPOND YES OR NO'/)
      GO TO 350
8060  WRITE (6,8065,ERR=8000)
8065  FORMAT (/1X,'NO NAMES IN AUTHOR DICTIONARY'/)
      GO TO 500
C
C             I/O FORMATTING
C
9000  FORMAT (/1X,'ENTER AUTHORS NAME - (DOE, J. J.) ',
     -       '- FORMAT MUST MATCH RECORD'/)
9010  FORMAT (A32)
9020  FORMAT (A1,I4)
9030  FORMAT (I4)
9040  FORMAT (A32,50I4)
9050  FORMAT (/1X,'ALTERNATE PRINT OPTION?'/)
9060  FORMAT (A4)
9070  FORMAT (3A32,A12,A100,A32,A20,6A16,A16)
      END
```

```
C
C
C
C
C                    K G S K E Y E X
C
C
C
      SUBROUTINE KGSKEYEX(ISTAT)
C
C     WRITTEN BY JUDITH CALABRESE - 12 AUGUST 1981
C
C     SUBROUTINE CALLED BY KGSEXTRT.
C
C     PROVIDES THE KEYWORD RETRIEVAL FUNCTION.
C
C             * FIELD *
C
C     A FIELD SEARCH SEARCHES THE KGSDCTNY FILE FOR THE
C     DIRECT ADDRESS OF ALL KGS RECORDS CONTAINING THE
C     ENTERED KEYWORDS.  ONLY THE KEYWORD FIELD OF THE
C     DATA BASE IS SEARCHED.
C     THIS MODE ALSO PROVIDES UNION(OR)/INTERSECTION(AND)
C     CAPABILITY FOR RECORD RETRIEVAL.
C
C             * STRING *
C
C     A STRING SEARCH SEARCHES THE TITLE FIELD OF ALL
C     RECORDS ON THE KGS DATA BASE FOR A MATCH WITH
C     ENTERED KEYWORDS.  THE KGS DATA BASE IS READ
C     SEQUENTIALLY.
C
C             * LOGICAL UNIT ASSIGNMENTS *
C
C     LU1-KGSINDEX DATA FILE
C     LU2-KGSKEY.DTA (KGS DICTIONARY FILE)
C     LU3-PRINT DEVICE
C     LU4-TEMPORARY SCRATCH FILE
C     LU5-KGS DATA BASE
C     LU6-COMMAND DEVICE
C
C
C
      CHARACTER*16,INKEY(6)
      CHARACTER*16,DCNKEY
      CHARACTER*32,INAUTH
      CHARACTER*1,INDEX(30)
      CHARACTER*1,HLDCHR(6)
      CHARACTER*1,IDUMMY
C
C             DATA ELEMENTS READ FROM KGS DATA BASE RECORD
C
      CHARACTER*32,AUTHOR(3)
      CHARACTER*12,PUBDAT
      CHARACTER*100,TITLE
      CHARACTER*32,PUBNAM
```

```
      CHARACTER*20,REFER
      CHARACTER*16,KEYWRD(6)
      CHARACTER*16,DISCPN
C
      INTEGER IDATE(3)
      INTEGER*2 RECNO(60),POINTR(30),ISTAT,ILINE,INKYCT
      INTEGER*2 ICELL,ISUB,IOVFLW,IPRTSW,IPRTCT,IDXCT
      INTEGER*2 ITIMES,ISCRCT,IOVFL1,IERRCT
      INTEGER*2 IPAGE,ISCREC,IERRSW
C
C
C             INITIALIZE VALUES
C
      INKYCT=0
      IDXCT=0
      ISUB=0
      ITIMES=0
      ISCRCT=0
      IPRTSW=0
      IPRTCT=0
      IERRCT=0
      IANDOR=0
      IPAGE=1
      INAUTH='
      IERRSW=0
      DO 10 I=1,6
      INKEY(I)='
10    CONTINUE
C
C             DATE ROUTINE
C
      CALL DATE(IDATE)
C
C             INPUT PARAMETERS
C             FOR FIELD OR STRING SEARCH
C
100   WRITE (6,9000,ERR=8000)
      READ (6,9010,ERR=8000) IRESP
      IF (IRESP.EQ.'    ') GO TO 7000
      IF (IRESP.EQ.'FIEL') GO TO 500
      ILINE=1
      IF (IRESP.NE.'STRI') GO TO 8020
C
C
C             *****   STRING SEARCH   *****
C
C
      CALL KGSTRING (ISTAT,IDATE)
      GO TO 9999
C
C             *****   FIELD SEARCH   *****
C
C
C             UNION OR INTERSECTION
C             PARAMETERS ENTERED BY USER
C
```

```
C             INTERSECTION - IANDOR=0
C             UNION        - IANDOR=1
C
C
500   WRITE (6,9020,ERR=8000)
      READ (6,9010,ERR=8000) IRESP
      IF (IRESP.EQ.'    ') GO TO 7000
      IF (IRESP.EQ.'UNIO') THEN
        IANDOR=1
      ELSE IF (IRESP.EQ.'OR  ') THEN
        IANDOR=1
      ELSE IF (IRESP.EQ.'INTE') THEN
        IANDOR=0
      ELSE IF (IRESP.EQ.'AND ') THEN
        IANDOR=0
      ELSE
        ILINE=2
        GO TO 8020
      ENDIF
C
C             ACCEPT KEY WORDS FROM USER
C
      WRITE (6,9100,ERR=8000)
      DO 520 I=1,6
      READ (6,9030,ERR=8000) INKEY(I)
      IF (INKEY(I).EQ.'                ') GO TO 540
      HLDCHR(I)=INKEY(I)
      INKYCT=INKYCT+1
520   CONTINUE
540   IF (INKYCT.EQ.0) GO TO 7000
C
C
C             BUILD INDEX ARRAY FROM INDEX FILE
C
      OPEN (1,IOSTAT=ISTAT,ERR=8010,FILE='KGSIDX.DTA',RECL=5,
     - SHARE='SRO')
      DO 560 I=1,30
      READ (1,9040,IOSTAT=ISTAT,ERR=8060,END=580) INDEX(I),POINTR(I)
      IDXCT=IDXCT+1
560   CONTINUE
580   CLOSE (1,IOSTAT=ISTAT,ERR=8030)
C
C             OPEN DICTIONARY AND SCRATCH FILES
C
      OPEN (2,IOSTAT=ISTAT,ERR=8010,FILE='KGSKEY.DTA',RECL=256,
     - SHARE='SRO',ACCESS='DIRECT')
      OPEN (4,IOSTAT=ISTAT,ERR=8010,RECL=5,SHARE='ERW',
     - STATUS='SCRATCH')
C
C             FIND START OF OVERFLOW RECORDS ON DICTIONARY
C
      READ (2,9120,IOSTAT=ISTAT,REC=1) IOVFLW
C             CHECK FOR END-OF-FILE CONDITION
      IF (ISTAT.EQ.31.OR.ISTAT.EQ.32) GO TO 8100
      IF (ISTAT.NE.0) GO TO 8060
C
```

```
C             BINARY SEARCH OF INDEX
C
600   ISUB=ISUB+1
      IF (ISUB.GT.INKYCT) GO TO 900
      IDUMMY=HLDCHR(ISUB)
      CALL SEARCH (INDEX,IDXCT,IDUMMY,ICELL)
      IREC=POINTR(ICELL)
      IF (ICELL.GT.0) GO TO 700
      IERRCT=IERRCT+1
      GO TO 740
C
C             FIND KEYWORD ON DICTIONARY
C
700   READ (2,9050,IOSTAT=ISTAT,REC=IREC)
     - DCNKEY,(RECNO(I),I=1,60)
C             CHECK FOR END-OF-FILE CONDITION
      IF (ISTAT.EQ.31.OR.ISTAT.EQ.32) GO TO 8070
      IF (ISTAT.NE.0) GO TO 8060
      IREC=IREC+1
      IF (DCNKEY.LT.INKEY(ISUB)) GO TO 700
      IF (DCNKEY.EQ.INKEY(ISUB)) GO TO 800
C
C
C             SEARCH OVERFLOW FOR KEYWORD
C
740   IOVFL1=IOVFLW
750   READ (2,9050,IOSTAT=ISTAT,REC=IOVFL1) DCNKEY,
     - (RECNO(I),I=1,60)
C             CHECK FOR END-OF-FILE CONDITION
      IF (ISTAT.EQ.31.OR.ISTAT.EQ.32) GO TO 8070
      IF (ISTAT.NE.0) GO TO 8040
      IOVFL1=IOVFL1+1
      IF (DCNKEY.EQ.INKEY(ISUB)) GO TO 800
      GO TO 750
C
C             CREATE SCRATCH FILE OF RECORD NUMBERS
C
C
C                  WRITE FIRST RECORD
C
800   IF (ISCRCT.NE.0) GO TO 830
      ITIMES=1
      DO 820 I=1,60
      IF (RECNO(I).EQ.0) GO TO 825
      WRITE (4,9060,IOSTAT=ISTAT,ERR=8080) RECNO(I),ITIMES
      ISCRCT=ISCRCT+1
820   CONTINUE
825   REWIND 4
      GO TO 600
C
C             WRITE ADDITIONAL RECORD NUMBERS
C             TO SCRATCH FILE
C
830   DO 860 I=1,60
      IF (RECNO(I).EQ.0) GO TO 870
840   READ (4,9060,IOSTAT=ISTAT,ERR=8060,END=850) ISCREC,ITIMES
```

```
      IF (RECNO(I).NE.ISCREC) GO TO 840
      BACKSPACE 4
      ITIMES=ITIMES+1
      WRITE (4,9060,IOSTAT=ISTAT,ERR=8000) ISCREC,ITIMES
      REWIND 4
      GO TO 860
850   ITIMES=1
      WRITE (4,9060,IOSTAT=ISTAT,ERR=8000) RECNO(I),ITIMES
      ISCRCT=ISCRCT+1
      REWIND 4
860   CONTINUE
870   REWIND 4
      GO TO 600
C
C             CLOSE DICTIONARY FILE
C             OPEN KGS DATA BASE
C             REWIND SCRATCH FILE
C
900   CLOSE (2,IOSTAT=ISTAT,ERR=8050)
      OPEN (5,IOSTAT=ISTAT,ERR=8010,FILE='KGS',RECL=372,SHARE='SRO',
     - ACCESS='DIRECT')
      REWIND 4
C
C             *****   RECORD RETRIEVAL FROM KGS DATA BASE   *****
C
C
C             ALTERNATE PRINT OPTION
C
C             HARD COPY OUTPUT - IPRTSW=1
C             OUTPUT ONLY ON TERMINAL - IPRTSW=0
C
      WRITE (6,9070,ERR=8000)
      READ (6,9010,ERR=8000) IRESP
      IF (IRESP.EQ.'NO  ') GO TO 1000
      IF (IRESP.NE.'YES ') GO TO 8050
      OPEN (3,IOSTAT=ISTAT,ERR=8010,FILE='PR2:')
      IPRTSW=1
C
C             UNION (OR)
C
1000  IF (IANDOR.EQ.0) GO TO 1200
1020  READ (4,9060,IOSTAT=ISTAT,ERR=8060,END=7000) ISCREC,ITIMES
      READ(5,9080,IOSTAT=ISTAT,ERR=8060,REC=ISCREC)
     - (AUTHOR(I),I=1,3),
     - PUBDAT,TITLE,PUBNAM,REFER,(KEYWRD(J),J=1,6),DISCPN
C
C             DISPLAY RECORD
C
      CALL KGSDSPLY(INKEY,INAUTH,IDATE,ISCREC,
     -              IANDOR,IPRTSW,IPAGE,IPRTCT,
     - AUTHOR,PUBDAT,TITLE,PUBNAM,REFER,KEYWRD,DISCPN)
C
      GO TO 1020
C
C             INTERSECTION (AND)
C
```

```
1200  IF (IERRCT.NE.0) INKYCT=INKYCT-IERRCT
1220  READ (4,9060,IOSTAT=ISTAT,ERR=8060,END=1250) ISCREC,ITIMES
      IF (ITIMES.NE.INKYCT) GO TO 1220
      READ(5,9080,IOSTAT=ISTAT,ERR=8060,REC=ISCREC)
     - (AUTHOR(I),I=1,3),
     - PUBDAT,TITLE,PUBNAM,REFER,(KEYWRD(J),J=1,6),DISCPN
      IERRSW=1
C
C             DISPLAY RECORD
C
      CALL KGSDSPLY(INKEY,INAUTH,IDATE,ISCREC,
     -              IANDOR,IPRTSW,IPAGE,IPRTCT,
     - AUTHOR,PUBDAT,TITLE,PUBNAM,REFER,KEYWRD,DISCPN)
      GO TO 1220
1250  IF (IERRSW.EQ.0) GO TO 8090
C
C             CLOSEOUT
C
7000  CLOSE (4,IOSTAT=ISTAT,ERR=8030)
      CLOSE (5,IOSTAT=ISTAT,ERR=8030)
      CLOSE (3,IOSTAT=ISTAT,ERR=8030)
      ISTAT=0
C
C             RETURN
C
9999  RETURN
C
C             *****   ERROR REPORTING AND FORMATTING   *****
C
8000  WRITE (6,8005,ERR=9999)
8005  FORMAT (/1X,'COMMAND DEVICE ERROR'/)
      ISTAT=1
      PAUSE
      GO TO 9999
8010  WRITE (6,8015,ERR=8000) ISTAT
8015  FORMAT (/1X,'ERROR ON OPENING FILE - IOSTAT = ',I4/)
      ISTAT=1
      PAUSE
      GO TO 9999
8020  WRITE (6,8025,ERR=8000)
8025  FORMAT (/1X,'INVALID RESPONSE - PLEASE ENTER AGAIN'/)
      GO TO (100,500),ILINE
8030  WRITE (6,8035,ERR=8000) ISTAT
8035  FORMAT (/1X,'ERROR ON CLOSING FILE - IOSTAT = ',I4/)
      ISTAT=1
      PAUSE
      GO TO 9999
8040  WRITE (6,8045,ERR=8000) INKEY(ISUB)
8045  FORMAT (/1X,'KEYWORD NOT IN DICTIONARY - ',A16/)
      REWIND 2
      GO TO 600
8050  WRITE (6,8055,ERR=8000)
8055  FORMAT (/1X,'PLEASE RESPOND YES OR NO'/)
      GO TO 900
8060  WRITE (6,8065,ERR=8000) ISTAT
8065  FORMAT (/1X,'ERROR ON READING FILE - ISTAT = ',I4)
```

```
      PAUSE
      ISTAT=1
      GO TO 9999
8070  WRITE (6,8075,ERR=8000) INKEY(ISUB)
8075  FORMAT (/1X,'KEYWORD NOT IN DICTIONARY - ',A16)
      IERRCT=IERRCT+1
      GO TO 600
8080  WRITE (6,8085,ERR=8000) ISTAT
8085  FORMAT (/1X,'ERROR ON WRITING FILE - IOSTAT = ',I4/)
      ISTAT=1
      PAUSE
      GO TO 9999
8090  WRITE (6,8095,ERR=8000)
8095  FORMAT (/1X,'***   NO FILES ON DATA BASE',
     -        ' CONTAINING ALL SELECTED KEYWORDS   ***'/)
      GO TO 7000
8100  WRITE (6,8105,ERR=8000)
8105  FORMAT (/1X,'NO RECORDS IN DICTIONARY'/)
      GO TO 7000
C
C     **********   I/O FORMATTING   **********
C
C
9000  FORMAT (/1X,'FIELD OR STRING SEARCH BY KEYWORD?')
9010  FORMAT (A4)
9020  FORMAT (/1X,'UNION (OR) OR INTERSECTION (AND)')
9030  FORMAT (A16)
9040  FORMAT (A1,I4)
9050  FORMAT (A16,60I4)
9060  FORMAT (I4,I1)
9070  FORMAT (/1X,'ALTERNATE PRINT OPTION?')
9080  FORMAT (3A32,A12,A100,A32,A20,6A16,A16)
9100  FORMAT (/1X,'ENTER KEYWORDS')
9120  FORMAT (I4)
      END
```

```
C
C
C
C
C                K G S T R I N G
C
C
C
      SUBROUTINE KGSTRING (ISTAT,IDATE)
C
C
C
C
C             LOGICAL UNIT ASSIGNMENTS
C
C     LU3-PRINT DEVICE
C     LU5-KGS DATA BASE
C     LU6-COMMAD DEVICE
C
C
      CHARACTER*32,INAUTH
      CHARACTER*16,INKEY(6)
      CHARACTER*16,INSTRG
      CHARACTER*16,FIELD
C
C             DATA ELEMENTS READ FROM KGS DATA BASE RECORD
C
      CHARACTER*32,AUTHOR(3)
      CHARACTER*12,PUBDAT
      CHARACTER*100,TITLE
      CHARACTER*32,PUBNAM
      CHARACTER*20,REFER
      CHARACTER*16,KEYWRD(6)
      CHARACTER*16,DISCPN
C
      INTEGER*2 INKYCT,IPRTSW,IPAGE,IPRTCT,IREC
C     INTEGER IDATE(3)
C
C             INITIALIZE VALUES
C
      INAUTH='
      INKYCT=0
      IREC=0
      IANDOR=2
      IPRTSW=0
      IPAGE=1
      IPRTCT=0
      ISTYP='
C
C             ALTERNATE PRINT OPTION
C
C             HARD COPY OUTPUT - IPRTSW=1
C             OUTPUT ONLY ON TERMINAL - IPRTSW=0
C
30    WRITE (6,9030,ERR=8000)
      READ (6,9040,ERR=8000) IRESP
```

```
      IF (IRESP.EQ.'NO  ') GO TO 40
      IF (IRESP.NE.'YES ') GO TO 8030
      OPEN (3,IOSTAT=ISTAT,ERR=8010,FILE='PR2:')
      IPRTSW=1
C
C             ACCEPT INPUT KEYS FROM USER
C
C
40    WRITE (6,9000,ERR=8000)
      DO 50 I=1,6
      READ (6,9010,ERR=8000)INKEY(I)
      IF (INKEY(I).EQ.'    ') GO TO 80
      INKYCT=INKYCT+1
50    CONTINUE
80    IF (INKYCT.EQ.0) GO TO 900
C
C             OPEN KGS DATA BASE
C
      OPEN (5,IOSTAT=ISTAT,ERR=8010,FILE='KGS',RECL=372,SHARE='SRO')
C
C             READ DATA BASE RECORDS
C
200   READ (5,9020,IOSTAT=ISTAT,ERR=8020,END=900)(AUTHOR(I),I=1,3),
     - PUBDAT,TITLE,PUBNAM,REFER,(KEYWRD(J),J=1,6),DISCPN
      IREC=IREC+1
C
C
C             CHECK FOR DELETED RECORD
C
      IF (AUTHOR(1).EQ.'DELEDELEDELEDELEDELEDELEDELEDELE') GO TO 200
C
C
C
C
C
C             PERFORM STRING SEARCH
C             ON TITLE FIELD
C
      DO 400 I=1,INKYCT
      INSTRG=INKEY(I)
      CALL STSRCH (TITLE,INSTRG,MATCH)
      IF (MATCH.EQ.1) GO TO 800
400   CONTINUE
C
C             PERFORM STRING SEARCH
C             ON KEY WORD FIELDS ON KGS DATA BASE
C
      DO 600 I=1,INKYCT
      INSTRG=INKEY(I)
      DO 650 J=1,6
      FIELD=KEYWRD(J)
      CALL STSRCH (FIELD,INSTRG,MATCH)
      IF (MATCH.EQ.1) GO TO 800
650   CONTINUE
600   CONTINUE
C
```

```
C             NO MATCH
C
      GO TO 200
C
C
C             MATCH -
C             DISPLAY RECORD
C
800   CALL KGSDSPLY (INKEY,INAUTH,IDATE,IREC,
     -        IANDOR,IPRTSW,IPAGE,IPRTCT,
     - AUTHOR,PUBDAT,TITLE,PUBNAM,REFER,KEYWRD,DISCPN)
      MATCH=0
      GO TO 200
C
C             CLOSEOUT
C
900   CLOSE (5,IOSTAT=ISTAT,ERR=8030)
      CLOSE (3,IOSTAT=ISTAT,ERR=8030)
      ISTAT=0
C
C             RETURN
C
9999  RETURN
C
C             ERROR REPORTING AND FORMATTING
C
8000  WRITE (6,8005,ERR=9999)
8005  FORMAT (/1X,'COMMAND DEVICE ERROR'/)
      ISTAT=1
      GO TO 9999
8010  WRITE (6,8015,ERR=8000) ISTAT
8015  FORMAT (/1X,'ERROR ON OPENING FILE - IOSTAT = ',I4/)
      GO TO 9999
8020  WRITE (6,8025,ERR=8000) ISTAT
8025  FORMAT (1X,'ERROR ON READING FILE - IOSTAT = ',I4)
      GO TO 9999
8030  WRITE (6,8035,ERR=8000)
8035  FORMAT (/1X,'PLEASE RESPOND YES OR NO'/)
      GO TO 30
C
C             I/O FORMATTING
C
9000  FORMAT (/1X,'ENTER KEY WORDS FOR SEARCH'/)
9010  FORMAT (A16)
9020  FORMAT (3A32,A12,A100,A32,A20,6A16,A16)
9030  FORMAT (/1X,'ALTERNATE PRINT OPTION?'/)
9040  FORMAT (A4)
      END
```

```
C
C
C
C
C             S T S R C H
C
C
C
C
C
      SUBROUTINE STSRCH (FIELD,INSTRG,MATCH)
C     WRITTEN BY JUDITH CALABRESE - 22 SEPTEMBER 1981
C
C
C     PERFORMS STRING SEARCH ON FIELD UP TO 100 CHARACTERS LONG
C     USING AN INPUT STRING UP TO 50 CHARACTERS LONG FOR COMPARISON.
C
C             FIELD  - FIELD TO BE SEARCHED
C             INSTRG - INPUT STRING SEARCHING FOR
C             MATCH  - 1 = MATCH
C                      0 = NOMATCH
C
C
      CHARACTER*100,FIELD
      CHARACTER*50,INSTRG
      CHARACTER*50,HLDSTR
      CHARACTER*1,HLDCAR(2)
      CHARACTER*2,HOLD(2)
C
C             INITIALIZE VALUES
C
      LENGTH=0
      MATCH=0
C
C             CALCULATE LENGTH OF INPUT STRING
C
      DO 20 I=1,50
      J=I+1
      HLDCAR(1)(1:)=INSTRG(I:I)
      HLDCAR(2)(1:)=INSTRG(J:J)
      IF (HLDCAR(1).EQ.'  '.AND.HLDCAR(2).EQ.'  ') GO TO 40
      LENGTH=LENGTH+1
20    CONTINUE
C
C             LOAD PORTION OF FIELD INTO HOLD AREA
C
40    DO 100 I=1,100
      LAST=I+(LENGTH-1)
      HLDSTR(1:)=FIELD(I:LAST)
      IF (HLDSTR.EQ.INSTRG) GO TO 200
      IF (HLDSTR.EQ.'  ') GO TO 9999
C
C
C             COMPARE INPUT STRING TO HOLD AREA
C
      ICTR=0
```

```
      DO 60 J=1,LENGTH
      K=J+1
      HOLD(1)(1:)=HLDSTR(J:K)
      HOLD(2)(1:)=INSTRG(J:K)
C
      IF (HOLD(1).NE.HOLD(2)) GO TO 60
      ICTR=ICTR+1
60    CONTINUE
      IF (ICTR.EQ.LENGTH) GO TO 200
100   CONTINUE
      GO TO 9999
C
C             MATCH
C
200   MATCH=1
C
C             RETURN
C
9999  RETURN
      END
```

```
C
C
C
C
C                A L L
C
C
C
C
      SUBROUTINE KGSALL(ISTAT)
C
C     WRITTEN BY JUDITH CALABRESE - 3 SEP 81
C
C     PRODUCES LISTING OF ALL RECORDS ON KGS DATA BASE
C     SORTED BY AUTHOR - DUPLICATE RECORDS (WITH MULTIPLE
C     AUTHORS) WILL APPEAR.
C
C     RECORDS IN DICTIONARY OVERFLOW WILL NOT BE SORTED.
C
C             * LOGICAL UNIT ASSIGNMENT *
C
C     LU2-KGS AUTHOR DICTIONARY
C     LU3-PRINT DEVICE
C     LU5-KGS DATA BASE
C     LU6-COMMAND DEVICE
C
      CHARACTER*32,DCNNAM
      CHARACTER*16,INKEY(6)
C
C             DATA ELEMENTS READ FROM KGS DATA BASE RECORD
C
      CHARACTER*32,AUTHOR(3)
      CHARACTER*12,PUBDAT
      CHARACTER*100,TITLE
      CHARACTER*32,PUBNAM
      CHARACTER*20,REFER
      CHARACTER*16,KEYWRD(6)
      CHARACTER*16,DISCPN
C
      INTEGER IDATE(3)
      INTEGER*2 RECNO(50),IREC,IPRTSW,IPAGE,IPRTCT,KREC
C
C
C             INITIALIZE VALUES
C
      IREC=1
      IPRTSW=1
      IPAGE=0
      IPRTCT=0
C
C             DATE ROUTINE
C
      CALL DATE (IDATE)
C
C             OPEN AUTHOR DICTIONARY
C             OPEN KGS DATA BASE
```

```
C             OPEN PRINT FILE
C
      OPEN (5,IOSTAT=ISTAT,ERR=8010,FILE='KGS',RECL=372,
     -     SHARE='SRO',ACCESS='DIRECT')
      OPEN (2,IOSTAT=ISTAT,ERR=8010,FILE='KGSAUTH.DTA',
     -     RECL=232,SHARE='SRO',ACCESS='DIRECT')
      OPEN (3,IOSTAT=ISTAT,ERR=8010,FILE='CAL:KGSPRT.DTA',
     -     RECL=132,SHARE='EWO',STATUS='RENEW')
C
C             READ AUTHOR DICTIONARY
C             TO DETERMINE START OF
C             OVERFLOW RECORDS
C
      READ (2,9000,IOSTAT=ISTAT,ERR=8020) IOVFLW
      IREC=IREC+1
C
C             READ AUTHOR DICTIONARY
C
100   READ (2,9010,IOSTAT=ISTAT,REC=IREC)
     -      DCNNAM,(RECNO(I),I=1,50)
      IREC=IREC+1
      IPRTCT=0
      IPAGE=IPAGE+1
C
C             CHECK FOR END-OF-FILE
C
      IF (ISTAT.EQ.31.OR.ISTAT.EQ.32) GO TO 500
      IF (ISTAT.NE.0) GO TO 8020
C
C             RETRIEVE RECORDS FROM KGS DATA BASE
C
      DO 200 I=1,50
      IF (RECNO(I).EQ.0) GO TO 100
      READ (5,9020,IOSTAT=ISTAT,ERR=8020,REC=RECNO(I))
     -      (AUTHOR(J),J=1,3),
     - PUBDAT,TITLE,PUBNAM,REFER,(KEYWRD(K),K=1,6),DISCPN
C
C
C             PRINT OUT RECORD
C
      KREC=RECNO(I)
      CALL KGSDSPLY (INKEY,DCNNAM,IDATE,KREC,
     -          IANDOR,IPRTSW,IPAGE,IPRTCT,
     - AUTHOR,PUBDAT,TITLE,PUBNAM,REFER,KEYWRD,DISCPN)
C
200   CONTINUE
C
      GO TO 100
C
C             CLOSEOUT
C
500   CLOSE (2,IOSTAT=ISTAT,ERR=8030)
      CLOSE (3,IOSTAT=ISTAT,ERR=8030)
      CLOSE (5,IOSTAT=ISTAT,ERR=8030)
      ISTAT=0
      WRITE (6,9030,ERR=8000)
```

```
C
C             RETURN
C
9999  RETURN
C
C             ERROR REPORTING AND FORMATTING
C
8000  WRITE (6,8005,ERR=9999)
8005  FORMAT (/1X,'COMMAND DEVICE ERROR'/)
      ISTAT=1
      PAUSE
      GO TO 9999
8010  WRITE (6,8015,ERR=8000) ISTAT
8015  FORMAT (/1X,'ERROR ON OPENING FILE - IOSTAT = ',I4)
      PAUSE
      GO TO 9999
8020  WRITE (6,8025,ERR=8000) ISTAT
8025  FORMAT (/1X,'ERROR ON READING FILE - IOSTAT = ',I4)
      PAUSE
      GO TO 9999
8030  WRITE (6,8035,ERR=8000) ISTAT
8035  FORMAT (/1X,'ERROR ON CLOSING FILE - IOSTAT = ',I4)
      PAUSE
      GO TO 9999
C
C             I/O FORMATTING
C
9000  FORMAT (I4)
9010  FORMAT (A32,50I4)
9020  FORMAT (3A32,A12,A100,A32,A20,6A16,A16)
9030  FORMAT (/1X,'*****   PRINT FILE - KGSPRT.DTA   *****'/)
      END
```

```
C
C
C
C
C                K G S D S P L Y
C
C
C
C
C
      SUBROUTINE KGSDSPLY(INKEY,INAUTH,IDATE,IREC,
     -                IANDOR,IPRTSW,IPAGE,IPRTCT,
     - AUTHOR,PUBDAT,TITLE,PUBNAM,REFER,KEYWRD,DISCPN)
C
C
C     WRITTEN BY JUDITH CALABRESE - 20 AUGUST 1981
C
C     DISPLAYS KGS RECORD ON CRT AND OPTIONALLY PRINTS HARD COPY
C
C             ARGUMENTS PASSED
C
C     INKEY  - ARRAY OF KEYWORDS INPUT BY USER (A16 FORMAT)
C     INAUTH - AUTHOR'S NAME INPUT BY USER
C     IDATE - TODAY'S DATE
C     IREC - RECORD NUMBER ON KGS DATA BASE
C
C     IANDOR - SWITCH SIGNIFYING UNION OR INTERSECTION
C              0=INTERSECTION
C              1=UNION
C              2=STRING SEARCH
C     IPRTSW - SWITCH INDICATING OPTIONAL PRINT
C              1=HARD COPY
C
C     IPAGE  - PAGE COUNTER
C     IPRTCT - RECORD COUNTER
C
C          ***   KGS DATA BASE FIELDS   ***
C     AUTHOR
C     PUBDAT
C     TITLE
C     PUBNAM
C     REFER
C     KEYWRD
C     DISCPN
C
C             LOGICAL UNIT ASSIGNMENTS
C
C     LU3-PRINTER (OPTIONAL)
C     LU6-COMMAND DEVICE
C
      CHARACTER*32,INAUTH
      CHARACTER*16,INKEY(6)
      CHARACTER*12,IEXTRT
      CHARACTER*50,TITLE1(2)
C
C             DATA ELEMENTS FROM KGS RECORD
```

```
C
      CHARACTER*32,AUTHOR(3)
      CHARACTER*12,PUBDAT
      CHARACTER*100,TITLE
      CHARACTER*32,PUBNAM
      CHARACTER*20,REFER
      CHARACTER*16,KEYWRD(6)
      CHARACTER*16,DISCPN
C
      INTEGER*2 IREC,IPRTSW,IPAGE,IPRTCT
      INTEGER IDATE(3)
C
C             INITIALIZE VALUES
C
      IF (IANDOR.EQ.0) IEXTRT='INTERSECTION'
      IF (IANDOR.EQ.1) IEXTRT='UNION       '
      IF (IANDOR.EQ.2) IEXTRT='STRING  SRCH'
C
C             SET CARRIAGE CONTROL
C             TO PAGE EJECT
C
      CALL CARCON (3,1)
C
C             DIVIDE TITLE FIELD
C
      TITLE1(1)(1:)=TITLE(1:50)
      TITLE1(2)(1:)=TITLE(51:100)
C
C             CHECK PRINT SWITCH FOR OPTIONAL HARD COPY
C
      IF (IPRTSW.NE.1) GO TO 200
C
C             HEADINGS OUTPUT TO PRINTER
C
      IF (IPRTCT.GT.0) GO TO 50
      WRITE (3,9000,ERR=8000) IDATE(2),IDATE(3),IDATE(1)
      WRITE (3,9010,ERR=8000)
      IF (INAUTH.NE.'                                ') THEN
         WRITE (3,9020,ERR=8000) INAUTH,IPAGE
      ELSE
         WRITE (3,9030,ERR=8000) (INKEY(I),I=1,3),IPAGE,
     -     (INKEY(J),J=4,6)
         WRITE (3,9040,ERR=8000) IEXTRT
      END IF
      WRITE (3,9010,ERR=8000)
C
C             RECORDS OUTPUT TO PRINTER
C
50    WRITE (3,9050,ERR=8000) AUTHOR(1),IREC
      WRITE (3,9060,ERR=8000) AUTHOR(2)
      WRITE (3,9070,ERR=8000) AUTHOR(3)
      WRITE (3,9140,ERR=8000) DISCPN
      WRITE (3,9080,ERR=8000) PUBDAT
      WRITE (3,9090,ERR=8000) (TITLE1(I),I=1,2)
      WRITE (3,9110,ERR=8000) PUBNAM,REFER
      WRITE (3,9120,ERR=8000) (KEYWRD(I),I=1,3)
```

```
      WRITE (3,9130,ERR=8000) (KEYWRD(I),I=4,6)
      WRITE (3,9010,ERR=8000)
C
C             RECORDS OUTPUT TO CRT
C
200    WRITE (6,9010,ERR=8000)
      IF (INAUTH.NE.'                                    ') THEN
         WRITE (6,9020,ERR=8000) INAUTH,IPAGE
      ELSE
         WRITE (6,9030,ERR=8000) (INKEY(I),I=1,3),IPAGE,
     -     (INKEY(J),J=4,6)
         WRITE (6,9040,ERR=8000) IEXTRT
      END IF
      WRITE (6,9010,ERR=8000)
      WRITE (6,9160,ERR=8000) AUTHOR(1),IREC
      WRITE (6,9060,ERR=8000) AUTHOR(2)
      WRITE (6,9070,ERR=8000) AUTHOR(3)
      WRITE (6,9140,ERR=8000) DISCPN
      WRITE (6,9080,ERR=8000) PUBDAT
      WRITE (6,9090,ERR=8000) (TITLE1(I),I=1,2)
      WRITE (6,9170,ERR=8000) PUBNAM
      WRITE (6,9180,ERR=8000) REFER
      WRITE (6,9120,ERR=8000) (KEYWRD(I),I=1,3)
      WRITE (6,9130,ERR=8000) (KEYWRD(I),I=4,6)
      WRITE (6,9010,ERR=8000)
C
C             INCREMENT AND SET COUNTERS
C
      IPRTCT=IPRTCT+1
      IF (IPRTCT.NE.3) GO TO 9999
      IPRTCT=0
      IPAGE=IPAGE+1
C
C             CLOSEOUT
C
9999  RETURN
C
C             ERROR REPORTING AND FORMATTING
C
8000  WRITE (6,8005,ERR=9999)
8005  FORMAT (1X,'COMMAND DEVICE ERROR')
      GO TO 9999
C
C             I/O FORMATTING
C
9000  FORMAT (1H1,1X///20X,'K G S   P U B L I C A T I O N S ',1X,
     - ' D A T A B A S E   R E T R I E V A L',8X,I2,'/',I2,'/',I2)
9010  FORMAT (1X,'- - - - - - - - - - - - - - - - - - - - -',1X,
     - '- - - - - - - - - - - - - - - - - - -',1X,
     - '- - - - - - - - - -'/)
9020  FORMAT (1X,'AUTHOR EXTRACTED WAS:  ',A32,1X,'PAGE ',I4)
9030  FORMAT (1X,'KEY WORD/S SELECTED WERE: ',3(A16,2X),5X,
     - 'PAGE ',I4,/28X,3(A16,2X))
9040  FORMAT (1X,'SEARCH CRITERIA USED: ',A12)
9050  FORMAT (1X,'AUTHOR:              ',A32,
     - 10X,'RECORD NO. ',I4)
```

```
9060  FORMAT (1X,'AUTHOR(2):           ',A32)
9070  FORMAT (1X,'AUTHOR(3):           ',A32)
9080  FORMAT (1X,'PUBLICATION DATE:    ',A12)
9090  FORMAT (1X,'TITLE:               ',A50/,22X,A50)
9110  FORMAT (1X,'PUBLICATION NAME:    ',A32,2X,
     - 'REFERENCE:  ',A20)
9120  FORMAT (1X,'- - KEY WORDS - - ',3(A16,4X))
9130  FORMAT (20X,3(A16,4X))
9140  FORMAT (1X,'FILE CODE:           ',A16)
9160  FORMAT (1X,'AUTHOR:              ',A32,
     - 5X,'RECORD NO. ',I4)
9170  FORMAT (1X,'PUBLICATION NAME:    ',A32)
9180  FORMAT (1X,'REFERENCE:           ',A20)
      END
```

```
C
C
C
C
C                S E A R C H
C
C
C
C
      SUBROUTINE SEARCH(INDEX,ICTIDX,IDUMMY,I)
C
C     WRITTEN BY JUDITH CALABRESE - 23 JULY 1981
C
C     PERFORMS BINARY SEARCH OF INDEX ARRAY
C
      CHARACTER*1,INDEX(30)
      CHARACTER*1,IDUMMY
      INTEGER*2 I,HI,LO,MID,ICTIDX
C
C             INITIALIZE VALUES
C
      IF (ICTIDX.LE.0) GO TO 9999
      HI=ICTIDX
      LO=1
C
C             CALCULATE MID POINT
C
10    MID=(HI+LO)/2
C
C             SEARCH
C
      IF (IDUMMY.EQ.INDEX(MID)) GO TO 50
      IF (IDUMMY.GT.INDEX(MID)) LO=MID
      IF (IDUMMY.LT.INDEX(MID)) HI=MID
C
C             CHECK FOR LAST 2 RECORDS
C
      IF ((HI-LO).EQ.1) GO TO 30
      GO TO 10
C
C             FINAL SEARCH
C
30    MID=HI
      IF (IDUMMY.EQ.INDEX(MID)) GO TO 50
      MID=LO
      IF (IDUMMY.EQ.INDEX(MID)) GO TO 50
C
C             NOT ON INDEX
C
      GO TO 8000
C
C              CLOSEOUT
C
50    I=MID
9999  RETURN
C
```

```
C             ERROR REPORTING
C
8000  WRITE (6,8005,ERR=9999) IDUMMY
8005  FORMAT (1X,'INPUT KEY NOT ON INDEX - ',A1)
      I=0
      GO TO 9999
      END
```

```
      SUBROUTINE KGSINFO (ISTAT)
C
C     WRITTEN BY JUDITH CALABRESE - 27 OCTOBER 1981
C
C     PROVIDES INFORMATION ON KEYWORDS AND AUTHORS
C     FROM KGS DICTIONARIES.
C
C     LISTS NAME AND NUMBER OF TIMES A KEYWORD OR
C     AUTHOR NAME APPEARS IN THE KGS DATA BASE.
C
C             LOGICAL UNITS
C
C     LU1 - KGSAUTH.DTA (KGS AUTHOR DICTIONARY)
C     LU2 - KGSKEY.DTA  (KGS KEYWORD DICTIONARY)
C     LU3 - PRINTER
C     LU6 - COMMAND DEVICE
C
C
      CHARACTER*32,AUTHOR(2)
      CHARACTER*16,KEYWRD(2)
C
      INTEGER IDATE(3)
      INTEGER*2 KEYREC(60),IOVFLW,ICOUNT1,ICOUNT2,IREC1,IREC2
      INTEGER*2 IBOTH,ILINE,IENDSW
C
C
C             INITIALIZE VALUES
C
      IBOTH=0
      IENDSW=0
C
C             OPEN PRINT DEVICE
C
      OPEN (3,IOSTAT=ISTAT,ERR=8020,FILE='PR2:')
C
C             DATE ROUTINE
C
      CALL DATE (IDATE)
C
C             USER ENTERS MODE
C
      WRITE (6,9000,ERR=8000)
20    WRITE (6,9010,ERR=8000)
      READ (6,9020,ERR=8000) IRESP
      IF (IRESP.EQ.'    ') GO TO 900
      IF (IRESP.EQ.'KEY ') GO TO 500
      IF (IRESP.EQ.'AUTH') GO TO 100
      IF (IRESP.EQ.'BOTH') THEN
        IBOTH=1
        GO TO 100
      END IF
      GO TO 8010
C
C             *****   AUTHOR MODE   *****
C
C             OPEN AUTHOR DICTIONARY
```

```
C
100   OPEN (1,IOSTAT=ISTAT,ERR=8020,FILE='KGSAUTH.DTA',RECL=232,
     -      SHARE='SRO',ACCESS='DIRECT')
C
C             FIND START OF OVERFLOW RECORDS
C
      READ (1,9030,IOSTAT=ISTAT,ERR=8030,REC=1) IOVFLW
      IREC1=2
C
C             WRITE REPORT HEADERS
C
110   WRITE (3,9040,ERR=8000) IDATE(2),IDATE(3),IDATE(1)
      WRITE (3,9050,ERR=8000)
      WRITE (3,9060,ERR=8000)
      ILINE=0
C
C             READ FIRST AND 40TH RECORD ON AUTHOR DICTIONARY
C              (40TH RECORD IS READ FOR 2-COLUMN PRINT)
C
120   READ (1,9080,IOSTAT=ISTAT,REC=IREC1) AUTHOR(1),
     -     (KEYREC(I),I=1,50)
C             CHECK FOR END-OF-FILE CONDITION
      IF (ISTAT.EQ.31.OR.ISTAT.EQ.32) GO TO 400
      IF (ISTAT.NE.0) GO TO 8030
      IREC1=IREC1+1
      ICOUNT1=0
      DO 140 I=1,50
      IF (KEYREC(I).EQ.0) GO TO 160
      ICOUNT1=ICOUNT1+1
140   CONTINUE
C
C
160   IF (IENDSW.EQ.1) GO TO 200
      IREC2=IREC1+39
      READ (1,9080,IOSTAT=ISTAT,REC=IREC2) AUTHOR(2),
     -     (KEYREC(I),I=1,50)
C             CHECK FOR END-OF-FILE CONDITION
      IF (ISTAT.EQ.31.OR.ISTAT.EQ.32) THEN
        IENDSW=1
        GO TO 200
      END IF
      IF (ISTAT.NE.0) GO TO 8030
      ICOUNT2=0
      DO 180 I=1,50
      IF (KEYREC(I).EQ.0) GO TO 200
      ICOUNT2=ICOUNT2+1
180   CONTINUE
C
C             WRITE OUT INFORMATION
C
200   IF (IENDSW.EQ.1) THEN
        WRITE (3,9070,ERR=8000) AUTHOR(1),ICOUNT1
      ELSE IF (IENDSW.EQ.0) THEN
        WRITE (3,9070,ERR=8000) AUTHOR(1),ICOUNT1,AUTHOR(2),ICOUNT2
      END IF
      ILINE=ILINE+1
```

```
      IF (ILINE.EQ.40) THEN
         IREC1=IREC1+40
         GO TO 110
      END IF
      GO TO 120
C
C             CLOSE AUTHOR DICTIONARY
C
400   CLOSE (1)
      IF (IBOTH.EQ.0) GO TO 900
C
C
C             *****   KEYWORD MODE   *****
C
C
C             OPEN KEYWRD DICTIONARY
C
500   OPEN (2,IOSTAT=ISTAT,ERR=8020,FILE='KGSKEY.DTA',RECL=256,
     -      SHARE='SRO',ACCESS='DIRECT')
C
C             FIND START OF OVERFLOW RECORDS
C
      READ (2,9030,IOSTAT=ISTAT,ERR=8030,REC=1) IOVFLW
      IREC1=2
      IENDSW=0
C
C             WRITE REPORT HEADERS
C
510   WRITE (3,9090,ERR=8000) IDATE(2),IDATE(3),IDATE(1)
      WRITE (3,9100,ERR=8000)
      WRITE (3,9060,ERR=8000)
      ILINE=0
C
C             READ FIRST AND 40TH RECORD ON KEYWRD DICTIONARY
C              (40TH RECORD IS READ FOR 2-COLUMN PRINT)
C
520   READ (2,9120,IOSTAT=ISTAT,REC=IREC1) KEYWRD(1),
     -     (KEYREC(I),I=1,60)
C             CHECK FOR END-OF-FILE CONDITION
      IF (ISTAT.EQ.31.OR.ISTAT.EQ.32) GO TO 800
      IF (ISTAT.NE.0) GO TO 8030
      IREC1=IREC1+1
      ICOUNT1=0
      DO 540 I=1,60
      IF (KEYREC(I).EQ.0) GO TO 560
      ICOUNT1=ICOUNT1+1
540   CONTINUE
C
C
560   IF (IENDSW.EQ.1) GO TO 600
      IREC2=IREC1+39
      READ (2,9120,IOSTAT=ISTAT,REC=IREC2) KEYWRD(2),
     -     (KEYREC(I),I=1,60)
C             CHECK FOR END-OF-FILE CONDITION
      IF (ISTAT.EQ.31.OR.ISTAT.EQ.32) THEN
        IENDSW=1
```

```
        GO TO 600
      END IF
      IF (ISTAT.NE.0) GO TO 8030
      ICOUNT2=0
      DO 580 I=1,60
      IF (KEYREC(I).EQ.0) GO TO 600
      ICOUNT2=ICOUNT2+1
580   CONTINUE
C
C             WRITE OUT INFORMATION
C
600   IF (IENDSW.EQ.1) THEN
        WRITE (3,9110,ERR=8000) KEYWRD(1),ICOUNT1
      ELSE IF (IENDSW.EQ.0) THEN
        WRITE (3,9110,ERR=8000) KEYWRD(1),ICOUNT1,KEYWRD(2),ICOUNT2
      END IF
      ILINE=ILINE+1
      IF (ILINE.EQ.40) THEN
         IREC1=IREC1+40
         GO TO 510
      END IF
      GO TO 520
C
C             CLOSE KEYWRD DICTIONARY
C
800   CLOSE (2)
C
C
C             CLOSEOUT
C
900   CLOSE (3)
      ISTAT=0
C
C             RETURN
C
9999  RETURN
C
C             ERROR REPORTING AND FORMATTING
C
8000  WRITE (6,8005,ERR=8000)
8005  FORMAT (/1X,'COMMAND DEVICE ERROR'/)
      ISTAT=1
      GO TO 9999
8010  WRITE (6,8015,ERR=8000)
8015  FORMAT (/1X,'INVALID RESPONSE - PLEASE ENTER AGAIN'/)
      GO TO 20
8020  WRITE (6,8025,ERR=8000) ISTAT
8025  FORMAT (/1X,'ERROR ON OPENING FILE - IOSTAT =',I4/)
      GO TO 9999
8030  WRITE (6,8035,ERR=8000) ISTAT
8035  FORMAT (/1X,'ERROR ON READING FILE - IOSTAT = ',I4/)
      GO TO 9999
C
C             I/O FORMATTING
C
9000  FORMAT (//////10X,'K G S   I N F O R M A T I O N   M O D E'///)
```

```
9010  FORMAT (/10X,'ENTER INFORMATION DESIRED',//15X,'AUTHOR',5X,
     -         'KEY',5X,'BOTH'//)
9020  FORMAT (A4)
9030  FORMAT (I4)
9040  FORMAT (1H1//////,10X,'KGS PUBLICATIONS DATA BASE',5X,
     -        'AUTHOR LIST',10X,I2,'/',I2,'/',I2//)
9050  FORMAT (10X,'AUTHOR',15X,'OCCURRENCES',25X,'AUTHOR',16X,
     -        'OCCURRENCES')
9060  FORMAT (1X,'-----------------------------------------',
     -        '-----------------------------------------',
     -        '----------'//)
9070  FORMAT (1X,A32,1X,I4,20X,A32,1X,I4)
9080  FORMAT (A32,50I4)
9090  FORMAT (1H1//////,10X,'KGS PUBLICATIONS DATA BASE',5X,
     -        'KEYWORD LIST',10X,I2,'/',I2,'/',I2//)
9100  FORMAT (10X,'KEYWORD',20X,'OCCURRENCES',15X,'KEYWORD',20X,
     -        'OCCURRENCES')
9110  FORMAT (5X,A16,17X,I4,18X,A16,15X,I4)
9120  FORMAT (A16,60I4)
      END
```

```
C     PROGRAM KGSRENEW
C
C     WRITTEN BY JUDITH CALABRESE - 21 OCTOBER 1981
C
C     REMOVES DELETED RECORDS FROM THE KGS DATA BASE
C
C             LOGICAL UNIT ASSIGNMENTS
C
C     LU1 - KGS DATA BASE (READ ONLY)
C     LU2 - KGSTEMP - OUTPUT FILE (WRITE ONLY)
C     LU6 - COMMAND DEVICE
C
C             KGS DATA ITEMS
C
      CHARACTER*32,AUTHOR(3)
      CHARACTER*12,PUBDAT
      CHARACTER*100,TITLE
      CHARACTER*32,PUBNAM
      CHARACTER*20,REFER
      CHARACTER*16,KEYWRD(6)
      CHARACTER*16,DISCPN
C
      INTEGER*2 INCT,OUTCT
C
C             INITIALIZE VALUES
C
      INCT=0
      OUTCT=0
C
C             OPEN FILES
C
      OPEN (1,IOSTAT=ISTAT,ERR=8010,FILE='KGS',RECL=372,SHARE='SRO')
C
      OPEN (2,IOSTAT=ISTAT,ERR=8010,FILE='KGSTEMP',RECL=372,
     -      SHARE='EWO',STATUS='RENEW')
C
C             READ KGS RECORDS
C
100   READ (1,9000,IOSTAT=ISTAT,ERR=8020,END=500)
     -     (AUTHOR(I),I=1,3),PUBDAT,TITLE,PUBNAM,REFER,
     -     (KEYWRD(J),J=1,6),DISCPN
      INCT=INCT+1
C
C             CHECK FOR DELETED RECORDS
C
      IF (AUTHOR(1).EQ.'DELEDELEDELEDELEDELEDELEDELEDELE') GO TO 100
C
C             WRITE OUT GOOD RECORD
C
      WRITE (2,9000,IOSTAT=ISTAT,ERR=8030)
     -     (AUTHOR(I),I=1,3),PUBDAT,TITLE,PUBNAM,REFER,
     -     (KEYWRD(J),J=1,6),DISCPN
      OUTCT=OUTCT+1
      GO TO 100
C
C             CLOSEOUT
```

```
500   WRITE (6,9010,ERR=8000) INCT,OUTCT
      CLOSE (1)
      CLOSE (2)
9999  STOP
C
C             ERROR REPORTING AND FORMATTING
C
8000  WRITE (6,8005,ERR=9999)
8005  FORMAT (/1X,'COMMAND DEVICE ERROR'/)
      GO TO 9999
8010  WRITE (6,8015,ERR=8000) ISTAT
8015  FORMAT (/1X,'ERROR ON OPENING FILE - IOSTAT = ',I4/)
      GO TO 9999
8020  WRITE (6,8025,ERR=8000) ISTAT
8025  FORMAT (/1X,'ERROR ON READING FILE - IOSTAT = ',I4/)
      GO TO 9999
8030  WRITE (6,8035,ERR=8000) ISTAT
8035  FORMAT (/1X,'ERROR ON WRITING FILE - IOSTAT = ',I4/)
      GO TO 9999
C
C             I/O FORMATTING
C
9000  FORMAT (3A32,A12,A100,A32,A20,6A16,A16)
9010  FORMAT (/1X,'INPUT RECORDS = ',I4,/1X,'OUTPUT RECORDS =',I4/)
      END
```

```
$BATCH
C     PROGRAM KGSDCNY
C
C     WRITTEN BY JUDITH CALABRESE - 20 OCTOBER 1981
C
C     MAIN PROGRAM OF THE KGS DICTIONARY MODE.
C
C     CALLS SUBROUTINES TO CREATE KEYWORD AND AUTHOR DICTIONARIES
C     AND INDEXES.
C
C             LOGICAL UNIT ASSIGNMENTS
C
C     LU3 -  PRINT DEVICE
C     LU6 - COMMAND DEVICE
C
C
C
C             HEADER
C
      WRITE (6,9000,ERR=8000)
C
C             OPEN PRINT FILE
C
      OPEN (3,IOSTAT=ISTAT,ERR=8020,FILE='PR2:')
C
C             CREATE DICTIONARIES
C                AND INDEXES
C
      CALL KGSDCKEY
      CALL KGSDCAUT
C
C             CLOSEOUT
C
9999  STOP
C
C             ERROR REPORTING AND FORMATTING
C
8000  WRITE (6,8005,ERR=9999)
8005  FORMAT (/1X,'COMMAND DEVICE ERROR'/)
      GO TO 9999
8020  WRITE (6,8025,ERR=8000) ISTAT
8025  FORMAT (/1X,'ERROR ON OPENING FILE - IOSTAT = ',I4/)
      GO TO 9999
C
C             I/O FORMATTING
C
9000  FORMAT (/10X,'K G S   D I C T I O N A R Y   M O D E'//,
     -         10X,'THIS ROUTINE CREATES BOTH THE AUTHOR AND'/,
     -         10X,'KEYWORD DICTIONARIES AND INDEXES'/,
     -         10X,'IT IS A LENGTHY PROCESS AND SHOULD BE'/
     -         10X,'RUN OVERNIGHT'//)
      END
C
C
C
C
```

```
C              K G S D C K E Y
C
C
C
C
      SUBROUTINE KGSDCKEY
C
C     WRITTEN BY JUDITH CALABRESE 14 JULY 1981
C
C     CREATES A SORTED DICTIONARY OF KEYWORDS FOR THE KGS DATA BASE.
C
C     >>>>>   LOGICAL UNIT ASSIGNMENT   <<<<<
C
C     LU1 - KGS DATA BASE (READ ONLY)
C     LU2 - KGSKEY.DTA OUTPUT KEYFILE
C     LU3 - PRINT FOR LIST OF KEYS
C     LU6 - COMMAND DEVICE
C
C
      CHARACTER*16,KEYNAM(800)
      CHARACTER*16,INAME(6)
      CHARACTER*16,SPACES
      INTEGER*2 KEYREC(800,60),HOLDCT,IREC
C
C             INITIALIZE VALUES
C
      SPACES='
      DO 20 I=1,800
      KEYNAM(I)=SPACES
      DO 10 J=1,60
      KEYREC(I,J)=0
10    CONTINUE
20    CONTINUE
      ICT=0
      IREC=0
C
C            BUILD ARRAY OF FIRST KEYWORDS ON FILE
C
      OPEN(1,IOSTAT=ISTAT,ERR=8010,FILE='KGS',RECL=372,SHARE='SRO')
100   READ (1,9000,IOSTAT=ISTAT,ERR=8020) (INAME(I),I=1,6)
      IREC=IREC+1
      DO 120 I=1,6
      IF (INAME(I).EQ.SPACES) GO TO 120
      KEYNAM(I)=INAME(I)
      KEYREC(I,1)=IREC
      ICT=ICT+1
120   CONTINUE
      IF (ICT.EQ.0) GO TO 100
C
C            SORT FIRST KEYWORDS
C
      IF (ICT.EQ.1) GO TO 200
      ICT2=ICT-1
140   IFLAG=0
      DO 160 I=1,ICT2
      IF (KEYNAM(I).LE.KEYNAM(I+1)) GO TO 160
```

```
      IFLAG=1
      INAME(1)=KEYNAM(I)
      KEYNAM(I)=KEYNAM(I+1)
      KEYNAM(I+1)=INAME(1)
160   CONTINUE
      IF (IFLAG.EQ.1) GO TO 140
C
C               READ KEYS AND SORT INTO KEY ARRAY
C
200   READ (1,9000,IOSTAT=ISTAT,ERR=8020,END=300)(INAME(I),I=1,6)
      IREC=IREC+1
      DO 280 I=1,6
      IF (INAME(I).EQ.SPACES) GO TO 280
      DO 260 J=1,ICT
      IF (KEYNAM(J).NE.INAME(I)) GO TO 220
C
C             INPUT KEY ALREADY ON ARRAY
C
      DO 210 K=1,60
      IF (KEYREC(J,K).NE.0) GO TO 210
      KEYREC(J,K)=IREC
      GO TO 280
210   CONTINUE
C
C              ADD INPUT KEY TO ARRAY - SORT
C
220   IF (KEYNAM(J).LT.INAME(I)) GO TO 260
      HOLDCT=ICT+1
240   KEYNAM(HOLDCT)=KEYNAM(HOLDCT-1)
      DO 250 K=1,60
      KEYREC(HOLDCT,K)=KEYREC(HOLDCT-1,K)
250   CONTINUE
      HOLDCT=HOLDCT-1
      IF (HOLDCT.GT.J) GO TO 240
      KEYNAM(J)=INAME(I)
      KEYREC(J,1)=IREC
      DO 258 K=2,60
      IF (KEYREC(J,K).EQ.0) GO TO 270
      KEYREC(J,K)=0
258   CONTINUE
      GO TO 270
260   CONTINUE
C
C             APPEND INPUT KEY TO ARRAY
C
      KEYNAM(ICT+1)=INAME(I)
      KEYREC(ICT+1,1)=IREC
270   ICT=ICT+1
280   CONTINUE
C
      GO TO 200
C
C             WRITE ARRAY TO KGSKEY.DTA
C
300   CLOSE (1,IOSTAT=ISTAT,ERR=8040)
      OPEN(2,IOSTAT=ISTAT,ERR=8010,FILE='KGSKEY.DTA',RECL=256,
```

```
     - STATUS='RENEW',SHARE='ERW')
       IOVFCT=ICT+1
       WRITE (2,9120,ERR=8000) IOVFCT
       DO 420 I=1,ICT
       WRITE (2,9060,ERR=8000) KEYNAM(I),(KEYREC(I,J),J=1,60)
       WRITE (3,9080,ERR=8000) KEYNAM(I)
420    CONTINUE
C
C             CLOSEOUT
C
       CALL KGSINDEX (0)
       WRITE (6,9100,ERR=8000) ICT
9999   RETURN
C
C              ERROR REPORTING AND FORMATING
C
8000   WRITE (6,8005,ERR=9999)
8005   FORMAT (1X,'COMMAND DEVICE ERROR')
       GO TO 9999
8010   WRITE (6,8015,ERR=8000)ISTAT
8015   FORMAT (1X,'ERROR ON OPENING FILE - IOSTAT = ',I4)
       GO TO 9999
8020   WRITE (6,8025,ERR=8000) ISTAT
8025   FORMAT (1X,'ERROR ON READING FILE - IOSTAT = ',I4)
       GO TO 9999
8040   WRITE (6,8045,ERR=8000) ISTAT
8045   FORMAT (1X,'ERROR ON CLOSING FILE - IOSTAT = ',I4)
       GO TO 9999
C
C              I/O FORMATING
C
9000   FORMAT (260X,6A16,16X)
9060   FORMAT (A16,60I4)
9080   FORMAT (1X,A16)
9100   FORMAT (1X,'KEYWORDS IN DICTIONARY = ',I8)
9120   FORMAT (I4)
       END
C
C
C
C              K G S D C A U T
C
C
C
C
C
C
       SUBROUTINE KGSDCAUT
C
C      WRITTEN BY JUDITH CALABRESE 14 JULY 1981
C
C      CREATES A SORTED DICTIONARY OF AUTHORS ON THE KGS DATA BASE.
C
C      >>>>>   LOGICAL UNIT ASSIGNMENT   <<<<<
C
C      LU1 - KGS DATA BASE (READ ONLY)
C      LU2 - KGSAUTH.DTA OUTPUT AUTHOR FILE
```

```
C     LU3 - PRINT FOR LIST OF AUTHORS
C     LU6 - COMMAND DEVICE
C
C
      CHARACTER*32,AUTHOR(1000)
      CHARACTER*32,INAUTH(3)
      CHARACTER*32,SPACES
      INTEGER*2 KEYREC(1000,50),HOLDCT,IREC
C
C             INITIALIZE VALUES
C
      SPACES='
      DO 20 I=1,1000
      AUTHOR(I)=SPACES
      DO 10 J=1,50
      KEYREC(I,J)=0
10    CONTINUE
20    CONTINUE
      ICT=0
      IREC=0
C
C            BUILD ARRAY OF FIRST KEYWORDS ON FILE
C
      OPEN(1,IOSTAT=ISTAT,ERR=8010,FILE='KGS',RECL=372,SHARE='SRO')
100   READ (1,9000,IOSTAT=ISTAT,ERR=8020) (INAUTH(I),I=1,3)
      IREC=IREC+1
      DO 120 I=1,3
      IF (INAUTH(I).EQ.SPACES) GO TO 120
      AUTHOR(I)=INAUTH(I)
      KEYREC(I,1)=IREC
      ICT=ICT+1
120   CONTINUE
      IF (ICT.EQ.0) GO TO 100
C
C            SORT FIRST KEYWORDS
C
      IF (ICT.EQ.1) GO TO 200
      ICT2=ICT-1
140   IFLAG=0
      DO 160 I=1,ICT2
      IF (AUTHOR(I).LE.AUTHOR(I+1)) GO TO 160
      IFLAG=1
      INAUTH(1)=AUTHOR(I)
      AUTHOR(I)=AUTHOR(I+1)
      AUTHOR(I+1)=INAUTH(1)
160   CONTINUE
      IF (IFLAG.EQ.1) GO TO 140
C
C              READ KEYS AND SORT INTO KEY ARRAY
C
200   READ (1,9000,IOSTAT=ISTAT,ERR=8020,END=300)(INAUTH(I),I=1,3)
      IREC=IREC+1
      DO 280 I=1,3
      IF (INAUTH(I).EQ.SPACES) GO TO 280
      IF (INAUTH(I).EQ.'DELEDELEDELEDELEDELEDELEDELEDELE') GO TO 280
      DO 260 J=1,ICT
```

```
      IF (AUTHOR(J).NE.INAUTH(I)) GO TO 220
C
C             INPUT KEY ALREADY ON ARRAY
C
      DO 210 K=1,50
      IF (KEYREC(J,K).NE.0) GO TO 210
      KEYREC(J,K)=IREC
      GO TO 280
210   CONTINUE
C
C              ADD INPUT KEY TO ARRAY - SORT
C
220   IF (AUTHOR(J).LT.INAUTH(I)) GO TO 260
      HOLDCT=ICT+1
240   AUTHOR(HOLDCT)=AUTHOR(HOLDCT-1)
      DO 250 K=1,50
      KEYREC(HOLDCT,K)=KEYREC(HOLDCT-1,K)
250   CONTINUE
      HOLDCT=HOLDCT-1
      IF (HOLDCT.GT.J) GO TO 240
      AUTHOR(J)=INAUTH(I)
      KEYREC(J,1)=IREC
      DO 258 K=2,50
      IF (KEYREC(J,K).EQ.0) GO TO 270
      KEYREC(J,K)=0
258   CONTINUE
      GO TO 270
260   CONTINUE
C
C             APPEND INPUT KEY TO ARRAY
C
      AUTHOR(ICT+1)=INAUTH(I)
      KEYREC(ICT+1,1)=IREC
270   ICT=ICT+1
280   CONTINUE
C
      GO TO 200
C
C             WRITE ARRAY TO KGSAUTH.DTA
C
300   CLOSE (1,IOSTAT=ISTAT,ERR=8040)
      OPEN(2,IOSTAT=ISTAT,ERR=8010,FILE='KGSAUTH.DTA',RECL=232,
     - STATUS='RENEW',SHARE='ERW')
      IOVFCT=ICT+1
      WRITE (2,9120,ERR=8000) IOVFCT
      WRITE (3,9130,ERR=8000)
      DO 420 I=1,ICT
      WRITE (2,9060,ERR=8000) AUTHOR(I),(KEYREC(I,J),J=1,50)
      WRITE (3,9080,ERR=8000) AUTHOR(I)
420   CONTINUE
C
C             CLOSEOUT
C
      CLOSE(3,IOSTAT=ISTAT,ERR=8040)
      CALL KGSINDEX (1)
      WRITE (6,9100,ERR=8000) ICT
```

```
9999  RETURN
C
C             ERROR REPORTING AND FORMATING
C
8000  WRITE (6,8005,ERR=9999)
8005  FORMAT (1X,'COMMAND DEVICE ERROR')
      GO TO 9999
8010  WRITE (6,8015,ERR=8000)ISTAT
8015  FORMAT (1X,'ERROR ON OPENING FILE - IOSTAT = ',I4)
      GO TO 9999
8020  WRITE (6,8025,ERR=8000) ISTAT
8025  FORMAT (1X,'ERROR ON READING FILE - IOSTAT = ',I4)
      GO TO 9999
8040  WRITE (6,8045,ERR=8000) ISTAT
8045  FORMAT (1X,'ERROR ON CLOSING FILE - IOSTAT = ',I4)
      GO TO 9999
C
C             I/O FORMATING
C
9000  FORMAT (3A32,276X)
9060  FORMAT (A32,50I4)
9080  FORMAT (1X,A32)
9100  FORMAT (1X,'AUTHORS IN DICTIONARY = ',I8)
9120  FORMAT (I4)
9130  FORMAT (1H1//1X)
      END
C
C
C
C
C
C             K G S I N D E X
C
C
C
C
C
      SUBROUTINE KGSINDEX (IFILE)
C
C     WRITTEN BY JUDITH CALABRESE - 15 JULY 81
C
C     CREATES THE INDEX USED TO ACCESS THE KGS DICTIONARY
C
C     >>>>>   LOGICAL UNIT ASSIGNMENTS   <<<<<
C
C     LU2 - KGSAUTH.DTA (OPENED IN MAIN)
C     LU4 - KGSIDXA.DTA
C     LU6 - COMMAND DEVICE
C
C
      CHARACTER*1,LETTER1
      CHARACTER*2,LETTER2
      INTEGER RECNO
C
C     INITIALIZE VALUES
C
```

```
      RECNO=1
      REWIND 2
C
C             OPEN INDEX FILE
C               IFILE = 0  -  OPEN KEYWORD INDEX
C               IFILE = 1  -  OPEN AUTHOR INDEX
C
      IF (IFILE.EQ.0)
     -OPEN (4,IOSTAT=ISTAT,ERR=8010,FILE='KGSIDX.DTA',RECL=5,
     - STATUS='RENEW',SHARE='ERW')
C
      IF (IFILE.EQ.1)
     -OPEN (4,IOSTAT=ISTAT,ERR=8010,FILE='KGSIDXA.DTA',RECL=5,
     - STATUS='RENEW',SHARE='ERW')
C
C     READ IN OVERFLOW POINTER FROM DICTIONARY
C
      IF (IFILE.EQ.0) READ (2,9000,ERR=8000) LETTER1
      IF (IFILE.EQ.1) READ (2,9010,ERR=8000) LETTER1
C
C     READ IN FIRST KEY FROM DICTIONARY
C
      IF (IFILE.EQ.0) READ (2,9000,IOSTAT=ISTAT,ERR=8020) LETTER1
      IF (IFILE.EQ.1) READ (2,9010,IOSTAT=ISTAT,ERR=8020) LETTER1
      RECNO=RECNO+1
      WRITE (4,9020,IOSTAT=ISTAT,ERR=8040) LETTER1,RECNO
C
C     READ REMAINING KEYS AND COMPARE
C
10    IF (IFILE.EQ.0)
     -READ (2,9000,IOSTAT=ISTAT,ERR=8020,END=9999) LETTER2
      IF (IFILE.EQ.1)
     -READ (2,9010,IOSTAT=ISTAT,ERR=8020,END=9999) LETTER2
      RECNO=RECNO+1
      IF (LETTER1.EQ.LETTER2) GO TO 10
      WRITE (4,9020,IOSTAT=ISTAT,ERR=8040) LETTER2,RECNO
      LETTER1=LETTER2
      GO TO 10
9999  RETURN
C
C>>>>>   ERROR REPORTING & FORMATING   <<<<<
C
8000  WRITE (6,8005,ERR=9999)
8005  FORMAT (1X,'COMMAND DEVICE ERROR')
      GO TO 9999
8010  WRITE (6,8015,ERR=8000) ISTAT
8015  FORMAT (1X,'ERROR ON OPENING FILE - ISTAT = ',I4)
      GO TO 9999
8020  WRITE (6,8025,ERR=8000) ISTAT
8025  FORMAT (1X,'ERROR ON READING FILE - ISTAT = ',I4)
      GO TO 9999
8040  WRITE (6,8045,ERR=8000) ISTAT
8045  FORMAT (1X,'ERROR ON WRITING FILE - ISTAT = ',I4)
      GO TO 9999
C
C     >>>>>   I/O FORMATTING   <<<<<
```

```
C
9000  FORMAT (A1,240X)
9010  FORMAT (A1,231X)
9020  FORMAT (A1,I4)
      END
$BEND
```

REFERENCES

MANUAL	REFERENCE NUMBER
OS/32 MT Operator's Reference Manual	29-574
MTM Terminal User's Reference Manual	B29-591
FORTRAN VII Level 1 Reference Manual	48-017R00
FORTRAN VII Level 1 User's Guide	S29-657R04

NBS-114A (REV. 2-8C)

U.S. DEPT. OF COMM. BIBLIOGRAPHIC DATA SHEET (See instructions)	1. PUBLICATION OR REPORT NO. NBS TN-1167	2. Performing Organ. Report No.	3. Publication Date October 1982

4. TITLE AND SUBTITLE

A Computer Data Base System for Indexing Research Papers

5. AUTHOR(S)

Judith T. Calabrese, Lawrence J. Kaetzel, Robert A. Glass, George R. Smith

6. PERFORMING ORGANIZATION (If joint or other than NBS, see instructions)

NATIONAL BUREAU OF STANDARDS
DEPARTMENT OF COMMERCE
WASHINGTON, D.C. 20234

7. Contract/Grant No.

8. Type of Report & Period Covered

Final

9. SPONSORING ORGANIZATION NAME AND COMPLETE ADDRESS (Street, City, State, ZIP)

Same as Item 6.

10. SUPPLEMENTARY NOTES

Supersedes NBS Technical Note 1123

☐ Document describes a computer program; SF-185, FIPS Software Summary, is attached.

11. ABSTRACT (A 200-word or less factual summary of most significant information. If document includes a significant bibliography or literature survey, mention it here)

This report represents a significant revision to NBS Technical Note 1123 published in 1980. In that report, the Kaetzel, Glass, Smith (KGS) data base system permitted users to index, edit, classify, and retrieve scientific research paper citations. During the past 15 months, the system was modified and enhanced. All programs are written in standard FORTRAN VII Level I programming language providing transportability among computer systems. Retrieval time has been greatly reduced by changing from a sequential access method to an indexed, directory look-up file structure which allows faster and more efficient random access. The file structure is machine independent. Because of the responsiveness of the extract mode, the one-key retrieval is unnecessary and has been deleted from the revised system. The keyword mode has been replaced by the information mode which provides statistics on authors and keywords. A file maintenance mode has been added to ensure data base integrity. The KGS system has been separated from the larger Publications Data Base and the select data base mode has been removed. Software has been tailored to meet KGS users' needs. Overall, the revised system is faster and uses resources more efficiently than the original data base.

12. KEY WORDS (Six to twelve entries; alphabetical order; capitalize only proper names; and separate key words by semicolons)

computer indexing; data base; directory look-up; information retrieval; interactive processing; random access.

13. AVAILABILITY

☒ Unlimited

☐ For Official Distribution. Do Not Release to NTIS

☒ Order From Superintendent of Documents, U.S. Government Printing Office, Washington, D.C. 20402.

☐ Order From National Technical Information Service (NTIS), Springfield, VA. 22161

14. NO. OF PRINTED PAGES

102

15. Price

$5.50

USCOMM-DC 6043-P80

☆U.S. GOVERNMENT PRINTING OFFICE: 1982-360-997/2284

NBS TECHNICAL PUBLICATIONS

PERIODICALS

JOURNAL OF RESEARCH—The Journal of Research of the National Bureau of Standards reports NBS research and development in those disciplines of the physical and engineering sciences in which the Bureau is active. These include physics, chemistry, engineering, mathematics, and computer sciences. Papers cover a broad range of subjects, with major emphasis on measurement methodology and the basic technology underlying standardization. Also included from time to time are survey articles on topics closely related to the Bureau's technical and scientific programs. As a special service to subscribers each issue contains complete citations to all recent Bureau publications in both NBS and non-NBS media. Issued six times a year. Annual subscription: domestic $18; foreign $22.50. Single copy, $4.25 domestic; $5.35 foreign.

NONPERIODICALS

Monographs—Major contributions to the technical literature on various subjects related to the Bureau's scientific and technical activities.

Handbooks—Recommended codes of engineering and industrial practice (including safety codes) developed in cooperation with interested industries, professional organizations, and regulatory bodies.

Special Publications—Include proceedings of conferences sponsored by NBS, NBS annual reports, and other special publications appropriate to this grouping such as wall charts, pocket cards, and bibliographies.

Applied Mathematics Series—Mathematical tables, manuals, and studies of special interest to physicists, engineers, chemists, biologists, mathematicians, computer programmers, and others engaged in scientific and technical work.

National Standard Reference Data Series—Provides quantitative data on the physical and chemical properties of materials, compiled from the world's literature and critically evaluated. Developed under a worldwide program coordinated by NBS under the authority of the National Standard Data Act (Public Law 90-396).

NOTE: The principal publication outlet for the foregoing data is the Journal of Physical and Chemical Reference Data (JPCRD) published quarterly for NBS by the American Chemical Society (ACS) and the American Institute of Physics (AIP). Subscriptions, reprints, and supplements available from ACS, 1155 Sixteenth St., NW, Washington, DC 20056.

Building Science Series—Disseminates technical information developed at the Bureau on building materials, components, systems, and whole structures. The series presents research results, test methods, and performance criteria related to the structural and environmental functions and the durability and safety characteristics of building elements and systems.

Technical Notes—Studies or reports which are complete in themselves but restrictive in their treatment of a subject. Analogous to monographs but not so comprehensive in scope or definitive in treatment of the subject area. Often serve as a vehicle for final reports of work performed at NBS under the sponsorship of other government agencies.

Voluntary Product Standards—Developed under procedures published by the Department of Commerce in Part 10, Title 15, of the Code of Federal Regulations. The standards establish nationally recognized requirements for products, and provide all concerned interests with a basis for common understanding of the characteristics of the products. NBS administers this program as a supplement to the activities of the private sector standardizing organizations.

Consumer Information Series—Practical information, based on NBS research and experience, covering areas of interest to the consumer. Easily understandable language and illustrations provide useful background knowledge for shopping in today's technological marketplace.

Order the **above** *NBS publications from: Superintendent of Documents, Government Printing Office, Washington, DC 20402.*

Order the **following** *NBS publications—FIPS and NBSIR's—from the National Technical Information Services, Springfield, VA 22161.*

Federal Information Processing Standards Publications (FIPS PUB)—Publications in this series collectively constitute the Federal Information Processing Standards Register. The Register serves as the official source of information in the Federal Government regarding standards issued by NBS pursuant to the Federal Property and Administrative Services Act of 1949 as amended, Public Law 89-306 (79 Stat. 1127), and as implemented by Executive Order 11717 (38 FR 12315, dated May 11, 1973) and Part 6 of Title 15 CFR (Code of Federal Regulations).

NBS Interagency Reports (NBSIR)—A special series of interim or final reports on work performed by NBS for outside sponsors (both government and non-government). In general, initial distribution is handled by the sponsor; public distribution is by the National Technical Information Services, Springfield, VA 22161, in paper copy or microfiche form.

U.S. Department of Commerce
National Bureau of Standards

Washington, D.C. 20234
Official Business
Penalty for Private Use $300

POSTAGE AND FEES PAID
U.S. DEPARTMENT OF COMMERCE
COM-215

THIRD CLASS
BULK RATE