NBS TECHNICAL NOTE 1123

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 — Thermodynamics and Molecular Science — 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] — Mechanical Engineering and Process Technology[2] — Building Technology — Fire Research — Consumer Product Technology — Field Methods.

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

Lawrence J. Kaetzel
Robert A. Glass
George R. Smith

Center for Building Technology
National Engineering Laboratory
National Bureau of Standards
Washington, D.C. 20234

U.S. DEPARTMENT OF COMMERCE, Philip M. Klutznick, Secretary
Luther H. Hodges, Jr., Deputy Secretary
Jordan J. Baruch, Assistant Secretary for Productivity, Technology and Innovation
NATIONAL BUREAU OF STANDARDS, Ernest Ambler, Director

Issued October 1980

National Bureau of Standards Technical Note 1123
Nat. Bur. Stand. (U.S.), Tech. Note 1123, 90 pages (Oct. 1980)
CODEN: NBTNAE

U.S. GOVERNMENT PRINTING OFFICE
WASHINGTON: 1980

For sale by the Superintendent of Documents, U.S. Government Printing Office, Washington, D.C. 20402
Price $4.25
(Add 25 percent for other than U.S. mailing)

A COMPUTER DATA BASE SYSTEM FOR INDEXING RESEARCH PAPERS

Lawrence J. Kaetzel
Robert A. Glass
George R. Smith

ABSTRACT

The Kaetzel-Glass-Smith data base system allows the indexing and retrieval of scientific research papers through the use of a minicomputer system in an interactive mode. Criteria are entered through the user's remote terminal which produces subsets of the data base in a report format as well as statistical summaries of data base elements.

Key words: Computer indexing; data base; information retrieval; interactive processing.

ACKNOWLEDGMENTS

The authors wish to thank the Occupational Safety and Health Administration for their support of this work. The KGS data base was developed in conjunction with OSHA's Office of Standards Development in order to be able to quickly retrieve references relevant to the development of a new standard on safety and hazard markings.

The authors also thank Mrs. Sharon Farinholt and Ms. Ulesia Gray for their efforts in the preparation of this manuscript and Dr. Fred Stahl, Mr. Frank Lerchen, and Mr. Dave Wise for their thoughtful suggestions.

TABLE OF CONTENTS

TABLE OF CONTENTS (Continued)

LIST OF TABLES

LIST OF FIGURES

LIST OF APPENDIX

1. INTRODUCTION

A set of computer programs written in the FORTRAN V Level I programming language has been developed which allows the indexing, classifying, retrieval, and editing of citations of research papers through the use of a computer terminal in a research laboratory or office. Complete listings of the application programs may be found in Appendix D.

The system is designed to help user's of medium and small computer systems organize, edit, file, and retrieve copies of research papers. A major advantage of implementing a system for personal copies of research papers, is that it allows the user to consider the references to form a research data base, and allows the user to search through them using new search strategies.

The current version of this data base system does not provide publication abstract capabilities. Enhancements to the system are planned during fiscal year 1981 and provisions for abstracting will be included.

This report describes the overall systems, its operation, and the computer program. 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.

A Perkin-Elmer[1] Model 7/32 minicomputer was used in the development and implementation of the system along with its associated peripherals and software routines. 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-half million bytes. Processing is performed using thirty-two bit word formats.

The computer system hosts a variety of peripheral devices to include rotating magnetic disk, magnetic tape, cassetter recorders, electrostatic printer/plotter and a medium speed line printer (see Fig. 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 four-hundred and fifty feet) are involved.

---

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

An asynchronous communications adapter is installed in 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 its remote devices.

2.2 Software

The software operating system installed in the computer is the "Perkin-Elmer OS/32MT" operating system. The system permits multi-user access to the computer's hardware and software resources. Priorities are assigned to a computer program during the development process, and determines the execution of the program with regard to peripheral and central processor availability.

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 V 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 Storage

Data elements (fields) of every KGS related-publication are stored in the data base in a prescribed format. Each publication represents one data base record, (see Fig. 2). Decisions relating to the representation of data within an element are the responsibility of the user. For example, a recommended format for the author name element could be; last name-comma-space-first initial-space-middle initial. The importance of using a consistent format is reflected in the retrieval of information when the "field" search criteria is employed.

Information retrieval is available in two formats (see Fig. 3). Optionally, the operator may have the information displayed on the CRT terminal or direct to a line printer device in the computer facility. Criteria used in selecting data base subsets are key word, author's name, and publication title. In addition, provisions are made to allow the entire data base to be displayed. Other features found in the retrieve mode are field versus character string search, and the union versus the intersection of a selected group of key words.

2.4 Data Base Sorting

The sequential ordering of the KGS data base is performed through the Perkin-Elmer Data System's "Sort-Merge" Software Program. The operator activates the sort program prior to entering the KGS data base system. Features of the sort routine include ascending or descending sequencing and sorting capability up to seven levels. Command files have been

---

* American Standard Code for Information Interchange.

prepared for the operator, which sort in ascending sequence by author and file code. By using this procedure, the entire data base is sorted and remains in that sequence until the next sort is performed.

3. MODES OF OPERATION

3.1 Data Base Supervisor Program (PDBSUP)

The function served by the data base supervisor program is to allow the operator to select from several data bases and to select the desired mode of operation. Refer to Figure 4 for software configuration. Program Modules are stored on disk and are loaded into memory only, when it is called by the operator (see Table 1). Upon completion of a run mode, control is returned to the supervisor program which idles waiting for another operator command. Error conditions which may occur during the execution of a run mode are reported on the operator's terminal, resulting in control returning to the supervisor program.

3.2 EDIT Mode (PDBED)

The user may edit the KGS data base by typing the command "EDIT". Execution of the EDIT mode begins by requesting one of the five EDIT sub-modes: 1) Search; 2) Update; 3) Append; 4) Delete; and 5) Return. All editing is performed by referencing the data base record number. Records are numbered from one to the number of the last record in the file. There is no maximum number of records except that which is dictated by disk storage capacity. Each data base record consists of fifteen lines (elements). Maximum line (element) sizes are preassigned for each record and are shown in Figure 2.

3.2.1 SEARCH Mode (PDBED)

During the process of editing the KGS data base, an identifier (record number) is used for accessing record information. The SEARCH mode is available to assist the operator in locating the desired record number. The program asks for the author's last name which is used in matching against each data base record. When a match is detected, the record number, author's name and publication title are displayed. An option to continue searching or terminate the search is available.

3.2.2 UPDATE Mode (PDBED)

The UPDATE mode is used to modify an element or elements in a data base record which is already in the data base. The program first asks for the record number, then displays the fifteen lines of text which are currently stored in the record. In addition to the text, the line or element number is affixed at the beginning of the line. These numbers, from one to fifteen are then used to identify the element (see Fig. 2).

By typing the line number, the program displays the current contents and asks for the updated information. The new text which has been entered is

then used to replace the old text. "Replace" is by line only, no character string capability is available. The program then asks for the next line number to be updated. When updating is completed, the operator types an asterisk, at which time the entire record is displayed and the user is given the option to save the updated record by typing "Y" or "N". A "Y" response will cause the information to be saved. An "N" response will cause the previously recorded text to be retained. The program then asks for the next record to be updated. When update is completed, the operator types an asterisk and another edit operation is requested.

3.2.3 APPEND Mode (PDBED)

The APPEND mode is used to splice new records into the data base. 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 operator enters the appropriate information for the entire record. The information is then displayed and the option to save or discard the information is given. A "YES" response will result in the record being stored on disk. A "NO" response will result in the request for the next record. By typing an asterisk, the APPEND mode is terminated and another mode is requested.

3.2.4 DELETE Mode (PDBED)

The nature of data base manipulation dictates that a method be provided for deleting unwanted records from the data base. The DELETE mode provides the operator with this capability. This process is achieved by marking the elements in the specified record with the characters "DELE". This suppresses the record for all future data base manipulation.

3.2.5 RETURN Mode (PDBED)

By typing "RETURN", the control of the system is returned to the data base supervisor program (PDBSUP).

3.3 KEY WORD Mode (PDBKEY)

The KEY WORD mode is designed to search each data base record and produce a sorted summary report of key words recorded in the key word fields. This program is activated in the run mode by typing the word "KEYS". An option is also provided which allows output to be displayed on the user CRT or on a hard copy device.

The report is generated in an alpha-numeric sorted order. In addition to the list of key words, the number of occurrences is indicated. A summary

---

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

is also presented which indicates the number of records in the data base, the number of "deleted" records, and the number of different key words.

3.4 SELECT DB Mode (PDBSUP)

The purpose of the SELECT DB mode is to allow the user to transfer control to another data base without terminating the KGS data base software package. This is useful if the user wishes to maintain publications on two or more areas of research. By typing "SELECT DB" the command "ENTER DATA BASE NAME" will be displayed. The data base name entered is subsequently used to activate the appropriate retrieval mode for data base manipulation. The valid data names are maintained in the supervisor program and are modified by deleting or inserting as needs arise.

3.5 ONE KEY Mode (PDBQK)

The purpose of the ONE KEY mode is to assist the data base operator in querying the data base for specific publication topics (key words). By typing a key word value when requested, the program searches the data base and displays those records (if any) which contained that key word. This is useful if a "snapshot" of the data base is desired, without displaying each data base record.

3.6 RETRIEVE Mode (KGSEX)

Information retrieval and display of the KGS data base is accomplished using the RETRIEVE mode. Various search criteria such as key word or author name may be entered and subsets of the data base extracted. Also included is the option for a field or character search of a data base record. A field search attempts to match the characters entered by the operator with the characters contained in the data base element. A string search will search the entire element for a match of the characters entered by the operator with any contiguous character match contained in the data base record element. The maximum number of characters in a string search is one hundred. In the case where the key word search option is used, each of the six key word fields plus the title field will be employed in searching. Optional printer hardcopy output is available. An option for the "UNION" or the "INTERSECTION" where more than one key word is also provided. In the UNION ('OR') condition, records must have all key words (maximum six) in either the title or key word field. In the INTERSECTION ('AND') condition a match on any of the entered key words will result in the record being retrieved.

When a string search criteria is employed the operator may enter the root of a key word which allow records to be retrieved having that root plus variations. For example:

Entering the characters "COL" might result in records having key words such as:

color
colour
colors
colours

4. OPERATING PROCEDURES

Each user is assigned a password, after the KGS system has been put into operation (as it has in the CBT Computer Center).

In order to activate and operate the KGS system, the user must have a terminal which is set in the proper telecommunications mode to allow capability with Perkin-Elmer 7/32 Computer System (or any host computer). For the CBT user, switches must be adjusted to reflect the conditions shown in Table 2.

4.1 Perkin-Elmer Computer System Signon Procedure

As discussed in Section 2.2, Computer Software, remote access to the Perkin-Elmer Computer System is provided through a Multi-Terminal Manager (MTM) subsystem. Two conditions must be met before a user may access the computer and activate the KGS data base. The first is that the MTM system must be active. A user may determine this by turning on the terminal, dialing the computer (if necessary), and depressing the "RETURN" key. If an asterisk is displayed by the computer, the computer system is available and the command to "SIGNON" may be entered. If no response is received, first ensure that all terminal telecommunication settings are correct (see Table 2). If so, contact the computer facility as to system availability. The second requirement of the operator is that the account number and password for the KGS data base be known. With these two conditions met, the operator may access the computer by typing the following command:

SIGNON ID,14,PDB (where ID equals a 1-8 character name for user identification purposes)

An acceptance of the "SIGNON" command will be acknowledged by the computer responding with an asterisk.

4.1.1 Operator Prompting

Operator interaction with the computer system may be described as having three levels. Each level of interaction is signified by the computer displaying a special character (prompt) in the left-most portion of the terminal display line. The presence of a prompt character notifies the operator that the system is available to accept operator commands. An asterisk (*) prompt represents the first or highest level and is the only level at which programs may be loaded. Additionally, other MTM commands may be entered. These include, "Display Time," (D T), "Display Users" (D U), "Cancel" (CA), etc. Refer to the Perkin-Elmer Multi-Terminal User's Manual for complete list of commands and their usage. The second

level, identified by a hyphen (-), acknowledges that a user task is loaded (such as the KGS data base). The third and final prompt is the greater than (>) sign. This is displayed when a program parameter is needed; such as the case when KGS data base is requesting a run mode. Depending on the system usage, situations will arise where a delay in the computer responding with a prompt will vary. Several factors determine the degree of delay in this response. These include the number of active terminal users, the run priority of the assigned tasks, and the amount of memory and disk resources being used. Normal response expected with the KGS system and five to six other programs running simultaneously is in the order of 1-5 seconds. This does not account for data base search time which may require longer periods of time due to data base size and the amount of information being retrieved.

4.2 Sorting

Sorting the KGS data base is performed by typing the following command:

KGSSORT

This will result in the sequencing of the data base in ascending order by author and file code within author.

4.3 Activating the KGS Data Base

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

The command to activate the data base is: PDB

This results in the program being loaded, and a message displayed requesting the operator to enter the desired satellite data base. Selection of the desired data base satellite results in a request that the operator enter the desired run mode.

4.4 Data Base Supervisor Program

When the data base system is activated, a message is displayed on the user terminal requesting the data base name. At this time, the operator should type "KGS". By doing so, the proper data base is assigned, and data record retrieval and manipulation may be performed in the proper format.

4.5 EDIT Mode

The user can edit the KGS data base by typing "EDIT". Then the program asks for one of 5 different run modes (see Table 4).

All of the editing is done through record numbers. The records in the data base are numbered from 1 to the total number of records in the file. Each record contains 15 lines. The SEARCH mode is useful in finding a record number, if the record number is not known (see Section 4.5.4).

4.5.1 APPEND Sub-Mode

The APPEND mode is used to add new records to the data base. If you type APPEND, the program asks for each of the 15 lines that constitutes a record. The user merely enters the appropriate text in the proper field, as requested by the KGS Editor. When the user is finished typing the 15 lines, the text is displayed on the screen. If the user wishes to save this record, he or she types "YES" to the question "SAVE OK?". If no, type "NO" (typing "NO" results in the just entered text being discarded). Then the program asks for the first line of the next record. If no more records are to be added, the user types an "*".

4.5.2 DELETE Sub-Mode

The DELETE mode is used to delete a record from the data base. The program asks for the record number of the record to delete. It then displays the author and title of that record and asks "DELETE OK?". The user types "YES" or "NO". The program then asks for the number of the next record to delete. When through deleting, enter an "*".

4.5.3 UPDATE Sub-Mode

The UPDATE mode is used to change a record already in the data base. The program asks for the record number of the record to be updated. The program then displays the 15 lines of text for that record. The lines are numbered.

The user types a line number to be changed and the program shows the current contents of that line and asks for the next line. The next line of text that the user types in replaces the old line. The program then asks for the next line to change. When finished changing lines of that record, type an "*".

The new record is displayed on the screen and the program asks "SAVE OK?". The user types "YES" or "NO". The program than asks for the next record to update. When finished updating, the user types an "*".

4.5.4 SEARCH Sub-Mode

The SEARCH mode is used to locate a record in the data base. The program asks for an author's name and searches the data base record author field for that author, employing a string search strategy. When a match is found, the record number of that record is displayed along with the author and title. The operator is then asked if the search process should

continue for that author. Type "NO" to end the search mode, and "YES" to find more records containing that author.

4.5.5 RETURN Mode

After entering an "*" to end an active edit mode, the user is asked for a new run mode. When finished editing, type "RETURN". The "RETURN" mode returns the user to the main supervisor program.

4.6 SELECT DB Mode

By typing "SELECT DB", the user may change data bases and then go into any of the modes of operation available in the KGS system.

4.7 KEY WORD Mode

By typing "KEY", the user may have an alphabetical list and number of occurrences of all key words in his data base (see Appendix B). An option for hardcopy or terminal display is provided.

4.8 ONE KEY Mode

By typing "ONE KEY", the user can query the system as to the existence of any number of references containing that key word, without actually retrieving the references.

4.9 RETRIEVE Mode

By typing the word "RETRIEVE," the KGS extract overlay module is activated. This program allows the user to select subsets of the data base by typing one of the following sequence of commands:

OPTIONAL PRINTER OUTPUT? (Yes or No)
(Answer Yes or No)

A "YES" response will cause the retrieved records to be directed to the line printer. A "NO" response will cause the retrieved records to be displayed at the user's terminal.

Field or String Search

Enter "FIELD" or "STRING"

Author or Key Word

If "AUTHOR" is typed, the user should respond with one of the following formats. Last name, First Initial, Middle Initial (for field search) or

"ALL" (for entire data base) or

Last name only (for string search)

If "KEY WORD" is typed, the user will be asked to enter the extract condition for more than one key word.

Key Word Union ("OR" condition) or Intersection? ("AND" condition)

Enter "UNION" or "INTERSECTION"

Continue?

Respond with "YES" for additional searches or "NO" for return to KGS data base main program segment.

4.10 STOP Mode

When finished operating the KGS data base system type: "STOP". The system will respond with the words "END OF TASK 0."

4.11 "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 "SIGNOFF" 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. Restrictions relating to 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 data base is twenty-three thousand bytes. This takes into consideration the program overlay techniques used in the program design and development process. Memory requirements may increase by three to four times if no program segmentation (overlay) is employed. Disk storage capacity necessary to operate the KGS system is determined by the number of records in the data base. The calculation necessary to determine this is 372 times the maximum number of records to be stored in the data base.

5.2 Software Installation

Applications programs for the KGS data base will most likely require minor modifications where a computer system other than an Perkin-Elmer 32 bit machine is employed. Changes may be necessary to substitute external or library subroutines. A list of these subroutines unique to the Perkin-Elmer computer is provided in Appendix E.

6. CONCLUSIONS

A description of an information retrieval system has been provided which can greatly enhance the development of research projects by allowing the researcher to interrogate computer-stored data bases.

The KGS system enhances the retrieval process by reducing the time spent by the researcher in identifying physical storage locations (file drawer, etc.) of research papers through the file code. A major advantage of implementing a system for personal copies of research papers is that it allows the user to consider all copies of research papers in the researcher's possession as forming a research data base. The user is assisted in searching through the reference articles using new search strategies. In this way, a given collection of papers can be used with greater efficiency to help support a variety of different research projects dealing with similar topics.

Implementation of the KGS data base system is within the realm of most minicomputer systems available today. In addition, the state of the art in microcomputers demonstrates their applicability where sufficient disk and memory capacity can be obtained.

Table 1. KGS Program Functions

MODULE NAME	PROGRAM FUNCTION
PDBSUP	Main program segment for data base selection and overlay caling.
PDBED	Provide insert, delete and update capability
PDBQK	Provides quick data base search existence of key word.
PDBKEY	Summarizes key word distribution across entire data base.
KGSEX	Retrieves and displays data base subsets based on selection criteria.

Table 2. Perkin-Elmer Remote Terminal Communications Protocol

ATTRIBUTE	SETTING
Baud Rate	110, 300, 1200, 2400, 9600 characters per second
Parity	Even
Number Stop Bits	1
Duplex	Full
Character Code Format	ASCII
Communications Interface	EIA RS-232
Number data bits	7

Table 3. Summary of SIGNON Procedures for the KGS Data Base System

1. Set switches (see Table 2)
2. Type "SIGNON ID,14,PDB"
3. When an asterisk appears type "PDB"
4. Select appropriate satellite data base
5. Select desired run mode.

Table 4. Available KGS Data Base Run Modes

MODE	SUB-MODE
EDIT	APPEND
	DELETE
	UPDATE
	SEARCH
	RETURN
RETRIEVE	
SELECT DB	
KEYS	
ONE KEY	
STOP	

Table 5. KGS Allowable Command Abbreviations*

COMMAND	ABBREVIATION OF COMMAND
EDIT	EDIT
APPEND	APP
DELETE	DE
UPDATE	UPD
SEARCH	SEA
RETURN	RET
RETRIEVE	RETR
SELECT DB	SELE
KEYS	KEYS
ONE KEY	ONE
STOP	STOP
AUTHOR	AUTH
UNION	UNI
INTERSECTION	INT
FIELD	FIEL
STRING	STRI

* NOTE: The user may elect to type the "complete command" or the abbreviation as desired.

Figure 1. CBT Perkin-Elmer Computer System Configuration

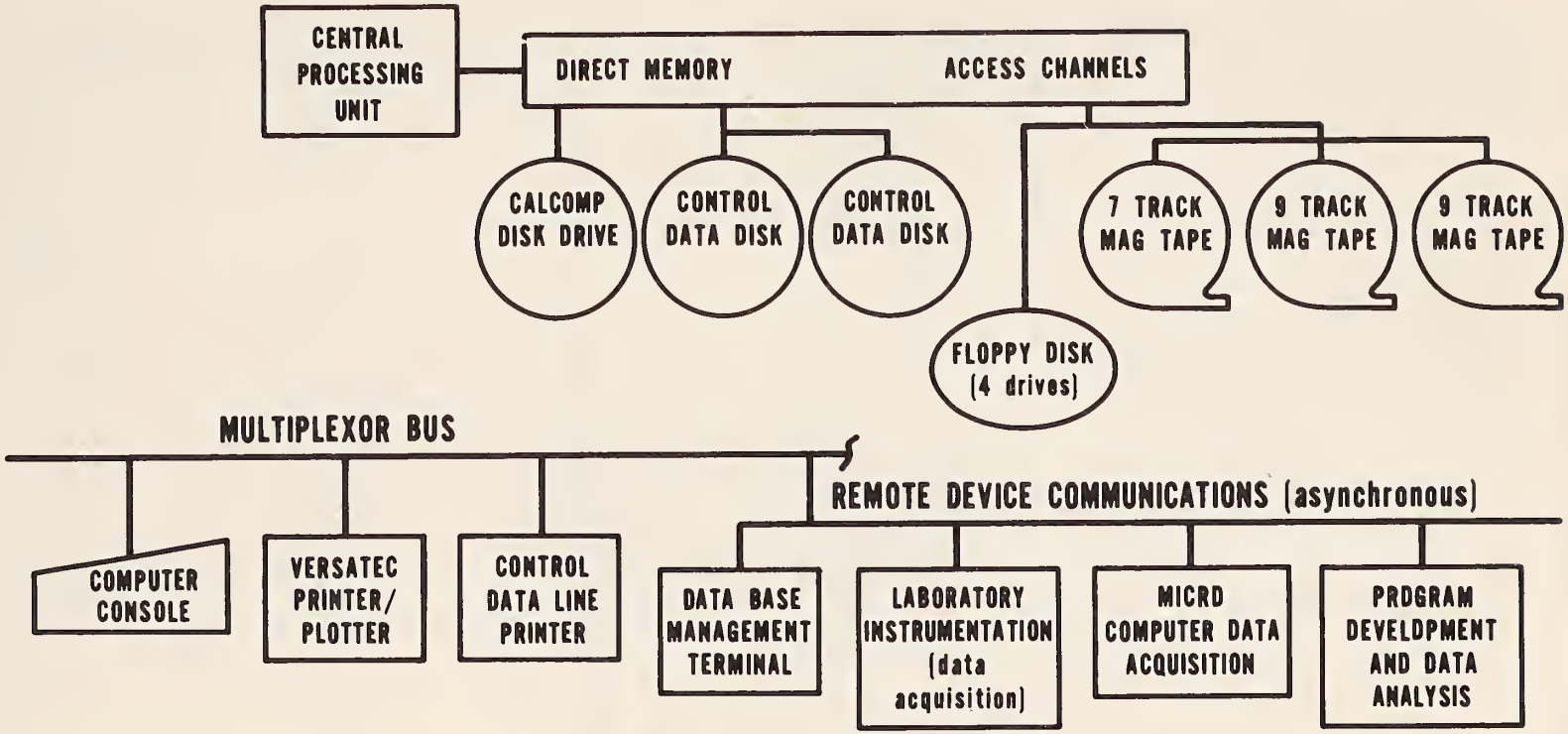


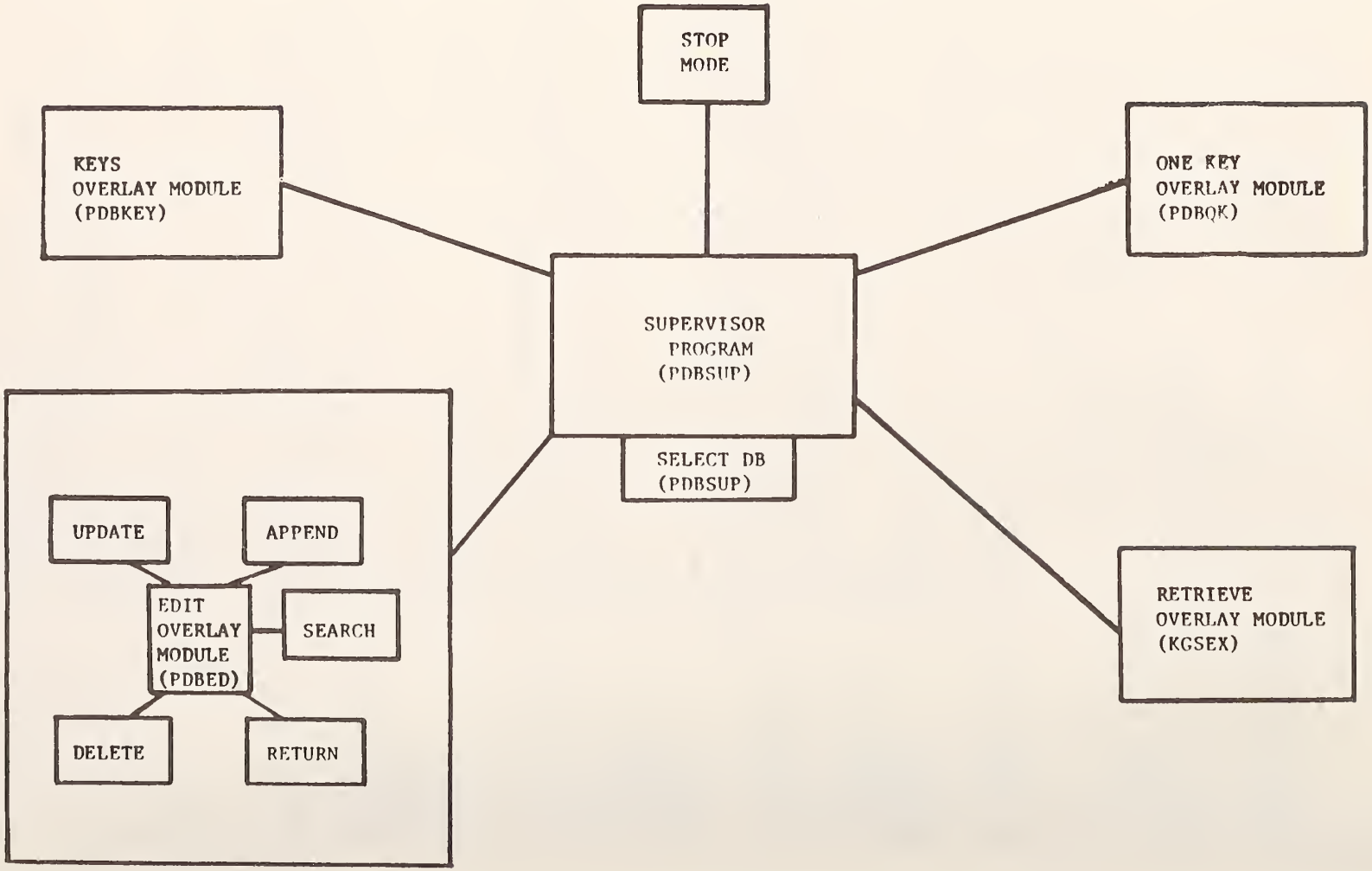
Figure 2. KGS Data Base Record Format

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)
Key Word 1 (16 characters)
Key Word 2 (16 characters)
Key Word 3 (16 characters)
Key Word 4 (16 characters)
Key Word 5 (16 characters)
Key Word 6 (16 characters)
Discipline (16 characters)

Figure 3. Sample KGS Retrieval

```
 KGS PUBLICATIONS DATABASE
 RECORD EXTRACT MODE       8/ 5/80   10:36:20

 OPTIONAL PRINTER OUTPUT? (YES OR NO)
NO
 ENTER EXTRACT CRITERIA (AUTHOR OR KEY WORD)
KEY WORD
      ENTER KEY WORD/S
 MAX 6 - 1 PER LINE - UP TO 16 CHAR EACH - "CR" TO STOP

HUE

- - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - -
KEY WORD/S SELECTED WERE: HUE

- - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - -
AUTHOR:              AKITA, M.                                RECORD NO.   14
AUTHOR(2):           GRAHAM, C. H.
AUTHOR(3):           HSIA, Y.
FILE CODE:           0232
PUBLICATION DATE:    0464
TITLE:               MAINTAINING AN ABSOLUTE HUE IN PRESENCE OF DIFFERENT
                      BACKGROUND COLORS
PUBLICATION NAME:    VISION RES.                        REFERENCE:  4,539-556
- - KEY WORDS - -  COLOR PERCEPTION    COLOR CONTRAST      ABSOLUTE HUE
                   BACKGROUND COLOR    LUMINANCE RATIO     BEZOLD-BRUCKE
- - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - -
AUTHOR:              BEARE, A. C.                             RECORD NO.   38
AUTHOR(2):
AUTHOR(3):
FILE CODE:           0266
PUBLICATION DATE:    0063
TITLE:               COLOR-NAME AS A FUNCTION OF WAVE-LENGTH

PUBLICATION NAME:    AMER. J. PSYCH.                    REFERENCE:  76,248-256
- - KEY WORDS - -  COLOR/NAME(S)       COLOR PERCEPTION    UNIQUE HUE(S)

- - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - -
AUTHOR:              BOYNTON, R. M.                           RECORD NO.   78
AUTHOR(2):           GORDON, J.
AUTHOR(3):
FILE CODE:           0048
PUBLICATION DATE:    0165
TITLE:               BEZOLD-BRUCKE HUE SHIFT MEASURED BY COLOR-NAMING TEC
                     HNIQUE
PUBLICATION NAME:    J. OPTICAL SOC. AMER.              REFERENCE:  55,1,78-86
- - KEY WORDS - -  BEZOLD-BRUCKE       HUE SHIFT           COLOR NAMING
                   UNIQUE HUES
- - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - -

AUTHOR:              BURNHAM, R. W.                           RECORD NO.  104
AUTHOR(2):           EVANS, R. M.
AUTHOR(3):           NEWHALL, S. M.
FILE CODE:           0008
PUBLICATION DATE:    0157
TITLE:               PREDICTION OF COLOR APPEARANCE WITH DIFFERENT ADAPTA
                     TION ILLUMINATIONS
PUBLICATION NAME:    J. OPTICAL SOC. AMER.              REFERENCE:  47,1,35-42
- - KEY WORDS - -  HUE SHIFT(S)        CHROM ADAPTATION    COLOR APPEARANCE
                   COLOR PERCEPTION
- - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - -
```

FIGURE 4: KGS PROGRAM MODULES



APPENDIX A: KGS SAMPLE SEARCH STRATEGY

```
PDB


 CBT PUBLICATIONS DATA BASE (REV 02.00-12/14/79)

          ENTER THE DATA BASE NAME YOU DESIRE
     - - FAA - - REHAB - - KGS - - AUTH740 - -
 - - AUTH741 - - AUTH742 - - AUTH743 - - AUTH744 - -
KGS

 ENTER DESIRED RUN MODE FOR
 K G S   DATA BASE

 - - EDIT - - RETRIEVE - - KEYS - - STOP - - SELECT DB - - ONE KEY - - ROSTER -
RETRIEVE
```

```
 KGS PUBLICATIONS DATABASE
 RECORD EXTRACT MODE        8/ 8/80    10:39:47

 OPTIONAL PRINTER OUTPUT? (YES OR NO)
NO
 ENTER EXTRACT CRITERIA (AUTHOR OR KEY WORD)
KEY WORD
      ENTER KEY WORD/S
 MAX 6 - 1 PER LINE - UP TO 16 CHAR EACH - "CR" TO STOP

RETINA
ANIMAL RESEARCH

 ENTER "UNION" (OR) OR "INTERSECTION" (AND) FOR SEARCH TYPE
INTERSECTION
 - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - -
 KEY WORD/S SELECTED WERE: RETINA            ANIMAL RESEARCH

 SEARCH CRITERIA USED: INT
 - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - -
 AUTHOR:              BARLOW, H. B.                          RECORD NO.   32
 AUTHOR(2):           LEVICK, W. R.
 AUTHOR(3):
 FILE CODE:           0197
 PUBLICATION DATE:    1976
 TITLE:               THRESHOLD SETTING BY THE SURROUND OF CAT RETINAL GAN
                      GLION CELLS
 PUBLICATION NAME:    J. PHYSIOL.                      REFERENCE:  259,737-757
 - - KEY WORDS - -  THRESHOLD          PHYSIOLOGY          RETINA
                    GANGLION CELL      ANIMAL RESEARCH
 - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - -
 AUTHOR:              DEMONASTERIO, F. M.                    RECORD NO.  179
 AUTHOR(2):           GOURAS, P.
 AUTHOR(3):           TOLHURST, D. J.
 FILE CODE:           0217
 PUBLICATION DATE:    1975
 TITLE:               CONCEALED COLOUR OPPONENCY IN GANGLION CELLS OF THE
                      RHESUS MONKEY RETINA
 PUBLICATION NAME:    J. PHYSIOL.                      REFERENCE:  251,217-229
 - - KEY WORDS - -  COLOR-OPPONENT     RETINA              RETINA CELLS
                    ANIMAL RESEARCH    GANGLIA
 - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - - -


 END OF FIELD SEARCH ON
 KEYWORDS:
    RETINA             ANIMAL RESEARCH


    2 RECORDS WERE EXTRACTED

 CONTINUE?
NO
```

```
 CBT PUBLICATIONS DATA BASE (REV 02.00-12/14/79)

 ENTER DESIRED RUN MODE FOR.
 K G S   DATA BASE

 - - EDIT - - RETRIEVE - - KEYS - - STOP - - SELECT DB - - ONE KEY - - ROSTER -
STOP
 * * *  END OF PROCESSING * * *
STOP
KGS     -END OF TASK CODE=   0    CPUTIME=1:49.539/3.169
SIGNOFF
ELAPSED TIME=2:36:34         CPUTIME=18:58.121/5:06.934
TIME OFF=08/08/80  10:52:00
```

APPENDIX B: SAMPLE KEY WORD LISTING

```
PDB


 CBT PUBLICATIONS DATA BASE (REV 02.00-12/14/79)

          ENTER THE DATA BASE NAME YOU DESIRE
     - - FAA - - REHAB - - KGS - - AUTH740 - -
 - - AUTH741 - - AUTH742 - - AUTH743 - - AUTH744 - -
KGS

 ENTER DESIRED RUN MODE FOR
 K G S   DATA BASE

 - - EDIT - - RETRIEVE - - KEYS - - STOP - - SELECT DB - - ONE KEY - - ROSTER -
KEYS

 BEGIN KEYWORD SORT
 OPTIONAL PRINTER OUTPUT? (YES OR NO)
NO
```

KGS PUBLICATIONS DATA BASE KEY WORD LIST 8/ 4/80

KEY WORD	OCCURRENCES
2-COLOR THRES	2
3-PIG COLOR VIS	1
ABSOLUTE HUE	1
ACHROMATIC	1
ACHROMATIC CHAN	1
ACHROMATIC COLOR	1
ADAP	1
ADAPTATION	10
AFFECT RESPONSE	1
AGE	1
ANIMAL BEHAVIOR	1
ANIMAL RESEARCH	3
APERIOD PATTERN	1
APP BRIGHTNESS	1
APPAR BRIGHTNESS	1
APPAR CONTRAST	1
APPARENT CONTR	1
ASYNCHRON FLASH	1
AVERAGE LUMINANC	1
BACKGROUND COLOR	1
BACKWARD MASKING	1
BAR WIDTH	1
BEZOLD-BRUCKE	5
BIBLIOGRAPHY	3
BINOCULAR VISION	1
BLACK BAR	1
BLOCHS LAW	1
BLUR	1
BORDER	3
BRIGHT CONSTANCY	1
BRIGHT ENHANCE	1
BRIGHT GRADIENTS	1
BRIGHT PERCEP	1
BRIGHT REVERSAL	1
BRIGHTNESS	13
BRIGHTNESS CONTR	1
BRIGHTNESS MATCH	1
CALIBRATION	1
CATHODE-RAY TUBE	1
CHLOROLABE	1
CHROM ADAPTATION	8
CHROM INDUCTION	1
CHROM ROD ACT	1
CHROM-CONTR SENS	1
CHROMATIC	1
CHROMATIC ADAPT	3
CHROMATIC BORDER	2
CHROMATIC CHAN	1
CHROMATIC RESP.	1
CHROMATIC THRES.	1
CHROMATICITY	2
CLOR PERCEPTION	1
CLOR RENDITION	1
COHERENCE	1
COL MEASUREMENT	1
COLOR	2
COLOR APPEAR.	1
COLOR APPEARANCE	6
COLOR ATTRIBUTE	1
COLOR BLINDNESS	3
COLOR CODING	9
COLOR CONRAST	1
COLOR CONSTANCY	2
COLOR CONTRAST	10
COLOR DEFECT	1
COLOR DEFECTS	1
COLOR DEFICIENCY	2
COLOR DIFFERENCE	2
COLOR DISCR	1
COLOR DISCRIM	3
COLOR DISCRIM.	1
COLOR DISCRIMINA	1
COLOR INDUCTION	1
COLOR MATCHING	2
COLOR MEMORY	1
COLOR NAMING	2
COLOR PERCEP.	1
COLOR PERCEPTION	17
COLOR RENDERING	1
COLOR RENDITION	1
COLOR RESPONSE	1
COLOR SHIFTS	1
COLOR SPEC	1
COLOR SYSTEM(S)	1
COLOR VIS	1
COLOR VISION	16
COLOR(S)	2
COLOR-OPPONENT	3
COLOR/NAME(S)	1
COLORED SIG LIGH	1

KGS PUBLICATIONS DATA BASE KEY WORD LIST 8/ 4/80

KEY WORD	OCCURRENCES
COLORED SIGNAL L	1
COLORIMETRIC COM	1
COLORIMETRY	1
COMPARATOR	2
COMPREHENSIBILIT	1
COMPREHENSION	1
CONCEN RECEP FLD	1
CONE VISION	1
CONE-TO-ROD RATI	1
CONES	1
CONSPICUITY	2
CONSTANCY	1
CONSTRAST	1
CONSUMER PRODUCT	2
CONTOUR	1
CONTOURS	1
CONTR DETECTION	1
CONTR REDUCTION	1
CONTR SENSITIVIT	1
CONTR THRESHOLD	2
CONTRAST	24
CONTRAST CONTOUR	1
CONTRAST HUES	1
CONTRAST SENSI	5
CONTRAST SENSIT	3
CONTRAST SENSIT.	2
CONTRAST THRESHO	1
CRAIK-CORNSWEET	1
CRITERIA	1
CROSS-MODALITIES	1
CULTURAL DIFF	1
CULTURAL DIFF.	1
CULTURAL FACTORS	2
DARK ADAPT	1
DARK ADAPT.	1
DARK ADAPTAION	1
DARK ADAPTATION	12
DEMOGRAPHIC	1
DESIGN	3
DETECTION	5
DETECTION THRES	1
DETERMINISTIC	1
DEUTERANOPIA	1
DICHROMATS	1
DIFF SENSITIVITY	1
DISCOMFORT GLARE	1
DISPLACEMENT	1
DISPLAY DENSITY	2
DISPLAYS	1
DISTINCTNESS	3
DUPLICITY THEORY	1
EBBINGHAUS	1
EDGE	2
EDGE DETECT MOVE	1
EDGE GRADIENT	2
EDGE(S)	1
EDIGE DETECTOR	1
EEG	1
EFFICIENCY	1
ELECTR POTENTIAL	1
ELECTRICA	1
ELECTRO-OPTIC	2
ELECTRONICS	1
EMERGENCY	1
EMERGENCY LGHTG	1
ENV. DESIGN	1
ENVIR DESIGN	1
ENVIRONMENTAL DE	1
EQU INNERVATION	1
EQU SPHERE ILLUM	1
EQUIPMENT	1
ERGONOMICS	2
ERYTHROLABE	1
ESI	1
ESI METER	1
ETHNIC DIFF	1
ETHNIC DIFF.	1
EVALUATION	1
EVOKED POTENTIAL	3
EVOKED RESPONSES	1
EXIT(S)	1
EYE	3
EYE EXAM.	1
EYE EXAMINATION	1
EYE MOVEMENT(S)	1
EYE MOVEMENTS	2
FATIGUE	3
FIELD TASK SIM.	1
FIELD(S)	1
FIG ORIENTATION	1

KGS PUBLICATIONS DATA BASE KEY WORD LIST 8/ 4/80

KEY WORD	OCCURRENCES	KEY WORD	OCCURRENCES
FILTER(S)	2	ILLUM LEVEL	1
FLASHING LIGHT	2	ILLUM. LEVEL	1
FLICKER	5	ILLUM. LEVEL(S)	1
FLICKER FUSION	2	ILLUMINANT	1
FLUORESCENT	1	ILLUMINATION	11
FLUORESCET COLOR	1	ILLUSIONS	1
FORM(S)	1	ILLUSTRATIONS	1
FOURIER ANALYSIS	2	IMAGE	1
FOVEA	3	IMAGE INTENS.	1
FOVEAL DIF THRES	1	IMAGE QUALITY	2
FOVEAL PERCEPT	1	IMAGE(S)	1
FOVEAL SENSIT	1	INCREMENT THRES	6
FOVEAL THRES	1	INCREMENT THRESH	1
FOVEAL VISION	1	INDUCED COLOR	1
GANGLIA	2	INFORMATION CHAN	1
GANGLION CELL	1	INFORMATION LOCA	1
GANZFELD	1	INSTRU LIGHTING	1
GAP-DET THRES	1	INSTRUMENT DESIG	1
GESTALT	1	INSTRUMENT PANEL	1
GLARE	1	INSTRUMENTATION	1
GOLDEN SECTION	2	INTERFER FILTER	1
GRADIENT	1	INTEROCULAR TRAN	1
GRATING	1	LATENCY	1
GRATING(S)	4	LATERAL INHIBIT	1
GRATINGS	5	LEGIBILITY	4
GRATINS	1	LIGGHT FLASH	1
GREED MECHANISMS	1	LIGHT	1
GREEN LIGHT	2	LIGHT CONTROL	1
HALLUCINATION(S)	1	LIGHT DISTRIBUT	1
HERINGS LAW	1	LIGHT FILTERS	1
HETEROCHRO PHOTO	1	LIGHT FLASH	3
HETEROCHROM BRI	1	LIGHT FLASH(ES)	5
HID LAMP(S)	1	LIGHT FLASHES	1
HIGHINTEN DIS	1	LIGHT QUALITY	2
HRR PLATES	1	LIGHT SIGNAL	1
HUE	4	LIGHT SOURCE(S)	2
HUE ESTIMATION	1	LIGHT SOURCES	2
HUE SHIFT	2	LIGHT SOURCES(S)	1
HUE SHIFT(S)	4	LIGHTING	3
HUE(S)	1	LIGHTNESS	3
HUMAN FACTORS	3	LIMULUS	1
HUMAN PERFORM	1	LIQUID CRYSTALS	2
IDENTIFICATION	1	LITERATURE SEAR	1
ILL LEVEL(S)	1	LITERATURE SEARC	1
ILLUM ENGINEER	1	LIVABILITY	1

KGS PUBLICATIONS DATA BASE KEY WORD LIST 8/ 4/80

KEY WORD	OCCURRENCES
LOGICAL TREES	1
LOW-PASS FILTER	1
LUM. EFFICIENCY	1
LUM. THRESHOLD(S	1
LUMIN EFF FUNC	1
LUMIN SUMMATION	1
LUMINANCE	5
LUMINANCE ADDIT	1
LUMINANCE GRAD	1
LUMINANCE LEVEL	1
LUMINANCE LEVELS	1
LUMINANCE LIMIT	1
LUMINANCE RATIO	3
LUMINANCE SENSI	1
LUMINANCE THRES	1
LUMINANCE THRESH	1
LUMINANCE(S)	1
LUMINOSITY FUNC	1
LUMINOUS EFF	1
MACH BAND	3
MACH BAND(S)	1
MACH BANDS	1
MANUAL	1
MASKING	2
MAXWELLIAN VIEW	1
MDS	1
MEASUREMENT	4
MEMORY COLOR	1
MEMORY COLOR(S)	1
MESOPIC	2
METHODLOGY	1
METHODOLOGY	5
METHODS	1
MIN DISTINCT BOR	1
MODEL	1
MODULATION SENSI	1
MODULATION TRANS	2
MONOCHROMATOR	1
MONOCULAR ADAP	1
MOOD(S)	1
MOOD-TONES	1
MOTION	1
MOTOR PERF.	1
MUELLER-LYER	1
MULTIDIM SCALING	1
MULTIDIMENSIONAL	1
MUNSELL	3
NEAR VISION	1
NIGHT VISION	3
NOISE	1
NON-UNIFORM ILLU	1
OPPONENT COLOR	2
OPPONENT COLORS	2
OPPONENT PROCESS	1
OPT ILLUM LEVEL	1
OPT. ILLUM. LEVE	1
OPTIC FILTER(S)	2
OPTICAL FILTER	1
OPTICAL ILLUSION	1
ORIENTATION	2
PAGE SIZE	1
PAINT(S)	2
PANEL DESIGN	2
PERCEIVED BRIGHT	1
PERCEP DISTORT	1
PERCEP ORIENTATI	1
PERCEPTAL DIST	1
PERCEPTION	2
PERIPHERAL VIS	1
PERIPHERAL VISIO	3
PHOTOBIOLOGY	2
PHOTOMETRY	1
PHOTORECEPTOR(S)	1
PHYSIOLOGICAL	1
PHYSIOLOGICAL RE	1
PHYSIOLOGY	11
PICTOGRAPHIC SYM	1
POWER SUPPLIES	1
PRE-ADAPT	1
PRE-ADAPTATION	5
PRE-STIMULATION	1
PRINT	1
PROB-SUMM MODEL	1
PRODUCT SAFETY	2
PRODUCT TESTING	2
PSYCHOPHYSICAL	1
PSYCHOPHYSICS	3
PSYCHOPHYSIOLOGI	1
PURITY	1
QUALITY CONTROL	1

KGS PUBLICATIONS DATA BASE KEY WORD LIST 8/ 4/80

KEY WORD	OCCURRENCES	KEY WORD	OCCURRENCES
QUANTITATIVE	2	SIGNAL LIGHT	1
QUAT MEASURE	1	SIGNAL LIGHT(S)	1
RATIO SCALING	2	SIGNAL RECOGN	1
REACTION TIME	3	SIGNAL(S)	2
READABILITY	1	SIMU CONTRAST	1
READING	1	SIMUL COLOR CONT	1
RECEPTIVE FIELD	1	SIMUL COLOR MATC	1
RECOGNITION	3	SIMUL COMPARISON	1
RECTANGLE(S)	2	SIMUL CONSTRAST	1
RED	1	SIMULTANEOUS COL	1
RED ADAPTATION	1	SINE WAVE GRAT	1
RED LIGHT	10	SINE-WAVE GRAD	1
RED MECHANISMS	1	SINE-WAVE GRAT	2
RED/GREEN	1	SINE-WAVE GRATIN	1
REDUNDANT CODING	1	SINE-WAVE TARGET	1
REFLEC MEASURE	1	SINUSOIDAL MOVE	1
REFLECTANCE	2	SIZE	2
REFLECTOMETER	1	SNELLEN	1
REFLUX TECHNIQUE	1	SODIUM LIGHTING	1
RESEARCH METHODS	1	SPATIAL EFFECT	1
RESOLUTION	1	SPATIAL FREQ	5
RETINA	4	SPATIO-TEMP INTE	1
RETINA CELLS	2	SPATIO-TEMPORAL	1
RETINAL BLUR	1	SPEC ENER CALIBR	1
RETINAL IMAGE	1	SPEC ENER DISTR	1
RETINAL IMAGE(S)	1	SPECTRAL ENERGY	1
RHODOSPIN	1	SPECTRAL SENSI	2
ROD VISION	3	SPECTRAL SENSIT	5
RODS	1	SQUARE-WAVE GRAT	1
SACCADIC EYE MOV	1	STABILIZED IMAGE	1
SAFETY	2	STABILIZED VIS	1
SAFETY COLOR(S)	1	STAIRCASE METHOD	1
SATURATION	8	STILES	1
SCALING	3	STIMU DURATION	3
SCOTOPIC	3	STIMUL DURATION	1
SECTOR DISC	2	STIMULUS DUR.	1
SEMANTIC DIFFERE	1	STOCHASTIC	1
SENS DEPRIVATION	1	SUBJECTIVE COLOR	3
SENTENCES	1	SUBJECTIVE EXPER	1
SHAPE CODING	1	SUCCESS COL MATC	1
SHARPENING	1	SUCCESS COMPAR	1
SHUTTER(S)	2	SUNSCREEN	1
SIGN DESIGN	2	SUPRATHRESHOLD	2
SIGN(S)	3	SYMBOL	1
SIGNAL DETECTION	1	SYMBOL(S)	2

KGS PUBLICATIONS DATA BASE KEY WORD LIST 8/ 4/80

KEY WORD	OCCURRENCES
SYMBOLS	2
TABLES	1
TASK COMPLEXITY	1
TEMP FREQUENCY	1
TEMPORAL INTERG	1
TERMINOLOGY	1
THERMAL COMFORT	1
THRES CONTRAST	1
THRES. CONTRAST	1
THRESHOLD	16
THRESHOLD SENSI	1
THRESHOLD(S)	9
THRRESHOLD	1
TRAFFIC SAFETY	1
TRAFFIC SAFTETY	1
TRAFFIC SIGN	1
TRAFFIC SIGN(S)	3
TRICHROMATIC	2
TUNGSTEN LIGHT	2
UNDERWATER	3
UNIQUE BLUE	2
UNIQUE GREEN	2
UNIQUE HUE	2
UNIQUE HUE(S)	2
UNIQUE HUES	3
UNIQUE RED	1
UNIQUE YELLOW	2
VALUE	1
VARIABILITY	1
VECTOR-MAG MODEL	1
VEILING LUMIN	1
VEP	1
VIEWING DISTANCE	1
VIS ACUITY	1
VIS DIFFICULTY	1
VIS INSPECTION	1
VIS PERCEPTION	1
VIS PERFORMANCE	9
VIS RECEPT CELL	1
VIS RESOLUTION	2
VIS RESPONSE SYS	1
VIS SEARCH PERF	1
VIS SENSITIVITY	1
VIS SIGNALLING	1
VIS TASK EVAL	1
VIS THRESHOLD	1
VIS THRESHOLDS	1
VIS. PERF.	1
VIS. SEARCH PERF	1
VIS. TASK EVAL.	1
VIS. THRESHOLD	2
VISIBILITY	12
VISIBILITY METER	4
VISION	2
VISUAL ACUITY	7
VISUAL CELLS	1
VISUAL COMFORT	1
VISUAL CRITERIA	1
VISUAL DESIGN	1
VISUAL DISPLAY	4
VISUAL DISPLAY(S	3
VISUAL ILLUSIONS	1
VISUAL INFORMATI	1
VISUAL PERCEPT	1
VISUAL PERF.	1
VISUAL PERFORM	1
VISUAL PERFORMAN	2
VISUAL PIGMENTS	1
VISUAL RECEPTOR	1
VISUAL RESPONSE	1
VISUAL SEARCH PE	1
VISUAL TASK	3
VISUAL THRESHOLD	3
VON KRIES	1
WAVEFORM(S)	1
WEBERS LAW	1
WHITE LIGHT	3
WHITE LIHT	1
WINDOWS	1
WRITING	1
X-CHROM LENS	2
XENON FLASHTUBE	1

```
      KGS       PUBLICATIONS DATA BASE     KEY WORD LIST      8/ 4/80


 END OF KEY WORDS

  647 RECORDS READ

    3 HAD BEEN DELETED

  532 KEY WORDS

  958 OCCURRENCES

 END OF KEYWORD SORT


 CBT PUBLICATIONS DATA BASE (REV 02.00-12/14/79)

 ENTER DESIRED RUN MODE FOR
 K G S   DATA BASE

- - EDIT - - RETRIEVE - - KEYS - - STOP - - SELECT DB - - ONE KEY - - ROSTER -
STOP
 * * *  END OF PROCESSING * * *
STOP
KGS     -END OF TASK CODE=   0    CPUTIME=2:18.508/3.936
SIGNOFF
ELAPSED TIME=1:03:32         CPUTIME=5:24.990/7.802
TIME OFF=08/04/80  15:09:50
```

APPENDIX C: PERKIN-ELMER OPERATING AND REFERENCE MANUALS

MANUAL	REFERENCE NUMBER
OS/32 MT Operator's Reference Manual	29-574
MTM Terminal User's Reference Manual	B29-591
Sort Merge Program Reference Manual	B29-489RØ1
FORTRAN V Level 1 Reference Manual	29-36ØRØ3
FORTRAN V Level 1 User's Guide	29-361RØ3
FORTRAN V 32-Bit Run Time Library	
Introduction	29-495
Language Extensions	29-497
Real Time Extension	29-499
Technical Descripton	29-5ØØ

APPENDIX D: KGS APPLICATIONS PROGRAM LISTING

```
C     PROGRAM PDBSUP.FTN
C     CBT PUBLICATIONS DATA BASE OVERLAY MODULE
C     SUPWERVISOR. DESIGNED TO CALL SUBROUTINES
C     FROM DISK FOR THE PURPOSE OF PROCESSING
C     QUERIES FROM A USER COMMAND DEVICE
C     WRITTEN BY LARRY KAETZEL - CBT - 03/02/79
C     >>>>>>>>>  LOGICAL UNIT ASSIGNMENTS  <<<<<<<<<<
C     LU 6 = COMMAND I/O DEVICE
C     LU 7 = OPTIONAL PRINT OUTPUT FILE
C     LU 3 = DATA BASE FILE
C     LU 4 = OVERLAY MODULE
C     MODIFIED 03/24/79 - 07:57
C     MODIFIED 08/10/79 - 18:13
C     MODIFIED 10/29/79 - 08:39
C     MODIFIED 10/30/79 - 08:28
C     MODIFIED 01/16/80 - 11:16
C     MODIFIED 06/03/80 - 09:30
      IMPLICIT INTEGER*2 (J-L)
      INTEGER IDATE(3),DBN(4),IMODE(5),BYTE1,BYTE2,KEY(4)
      INTEGER AUTDIV(5)
      DATA AUTDIV/'740 ','741 ','742 ','743 ','744 '/
      ISWH=0
      BYTE2=BYTE2
      DBN(1)='CAL:'
      CALL DATE(IDATE)
30    WRITE(6,1000,ERR=30)
      IF (ISWH.NE.0) GO TO 70
55    WRITE(6,1001,ERR=55)
60    READ(6,1013,ERR=800) (DBN(I),I=2,4)
      IF (DBN(2).EQ.'REHA') GO TO 70
      IF (DBN(2).EQ.'KGS ') GO TO 70
      IF (DBN(2).EQ.'AUTH') GO TO 70
      IF (DBN(2).EQ.'FAA ') GO TO 70
      WRITE(6,1003,ERR=800)
      GO TO 60
70    ISWH=1
      IF (DBN(2).EQ.'REHA') WRITE(6,1004,ERR=800)
      IF (DBN(2).EQ.'KGS ') WRITE(6,1005,ERR=800)
      IF (DBN(2).EQ.'FAA ') WRITE(6,1014,ERR=800)
      DO 65 I=1,5
      IF (DBN(3).EQ.AUTDIV(I)) GO TO 67
65    CONTINUE
      GO TO 68
67    IF (DBN(2).EQ.'AUTH') WRITE(6,1006,ERR=800)
68    WRITE(6,1007,ERR=800)
90    READ(6,1002,ERR=800) IMODE
      IF (IMODE(1).EQ.'SELE') GO TO 55
      IF(IMODE(1) .EQ. 'ONE ')GO TO 350
      IF (IMODE(1).EQ.'RETR'.AND.DBN(2).EQ.'REHA')
     -  GO TO 500
      IF (IMODE(1).EQ.'RETR'.AND.DBN(2).EQ.'KGS ')
     -  GO TO 530
```

```
      IF (IMODE(1).EQ.'RETR'.AND.DBN(2).EQ.'AUTH')
     -  GO TO 560
      IF (IMODE(1).EQ.'RETR'.AND.DBN(2).EQ.'FAA ')
     - GO TO 590
      IF (IMODE(1).EQ.'EDIT'.AND.DBN(2).EQ.'AUTH') GO TO 620
      IF (IMODE(1).EQ.'KEYS') GO TO 650
      IF (IMODE(1).EQ.'STOP') GO TO 700
      IF (IMODE(1).EQ.'EDIT') GO TO 600
      IF (IMODE(1).EQ.'ROST') GO TO 675
      CALL ILBYTE(BYTE1,IMODE,0)
      IF(BYTE1 .EQ. Y'0000002A')GO TO 400
      WRITE(6,1012,ERR=800)
      GO TO 90
C     >>>>>>>  ONE KEY  <<<<<<<
350   WRITE(6,351,ERR = 800)
351   FORMAT('0WHICH KEY WORD?')
      READ(6,360)KEY
360   FORMAT(4A4)
      GO TO 411
400   DO 405 I = 1,4
        IMODE(I) = ISHFT(IMODE(I),8)
        CALL ILBYTE(BYTE2,IMODE(I + 1),0)
        CALL ISBYTE(BYTE2,IMODE(I),3)
405     CONTINUE
      DO 410 I = 1,4
410     KEY(I) = IMODE(I)
411   CALL OPENW(4,'CAL:PDBQK.OVY   ',0,0,0,ISTAT)
      IF (ISTAT.NE.0) GO TO 810
      CALL IFETCH('PDBQK   ',4,4,ISTAT)
      IF (ISTAT.NE.0) GO TO 820
      CALL CLOSE(4,ISTAT)
      CALL PDBQK(KEY,DBN)
      GO TO 30
C     >>>>>>>  REHAB RETRIEVE MODE  <<<<<<<
500   CALL OPENW(4,'CAL:REHAEX.OVY  ',0,0,0,ISTAT)
      IF (ISTAT.NE.0) GO TO 810
      CALL IFETCH('REHAEX  ',4,4,ISTAT)
      IF (ISTAT.NE.0) GO TO 820
      CALL CLOSE(4,ISTAT)
      CALL REHAEX
      GO TO 30
C     >>>>>>>>>>  KGS RETRIEVE MODE  <<<<<<<<<<
530   CALL OPENW(4,'CAL:KGSEX.OVY   ',0,0,0,ISTAT)
      IF (ISTAT.NE.0) GO TO 810
      CALL IFETCH('KGSEX   ',4,4,ISTAT)
      IF (ISTAT.NE.0) GO TO 820
      CALL CLOSE(4,ISTAT)
      CALL KGSEX
      GO TO 30
C     >>>>>>>>>>  AUTHOR RETRIEVE MODE  <<<<<<<<<<
560   CALL OPENW(4,'CAL:AUTHEX.OVY  ',0,0,0,ISTAT)
      IF (ISTAT.NE.0) GO TO 810
```

```
      CALL IFETCH('AUTHEX  ',4,4,ISTAT)
      IF (ISTAT.NE.0) GO TO 820
      CALL AUTHEX
      GO TO 30
C     >>>>>>>  FAA RETRIEVE MODE  <<<<<<<<
590   CALL OPENW(4,'CAL:FAAEX.OVY   ',0,0,0,ISTAT)
      IF (ISTAT.NE.0) GO TO 810
      CALL IFETCH('FAAEX   ',4,4,ISTAT)
      IF (ISTAT.NE.0) GO TO 820
      CALL CLOSE(4,ISTAT)
      CALL FAAEX
      GO TO 30
C     >>>>>>>>>>  EDIT MODE  <<<<<<<<<<
600   CALL OPENW(4,'CAL:PDBED.OVY  ',0,0,0,ISTAT)
      IF (ISTAT.NE.0) GO TO 810
      CALL IFETCH('PDBED  ',4,4,ISTAT)
      IF (ISTAT.NE.0) GO TO 820
      CALL CLOSE(4,ISTAT)
      CALL PDBED(DBN)
      CALL CLOSE(3,ISTAT)
      GO TO 30
C     >>>>>>>>>>  EDIT  AUTHOR MODE  <<<<<<<<<<
620   CALL OPENW(4,'CAL:PDBED2.OVY  ',0,0,0,ISTAT)
      IF (ISTAT.NE.0) GO TO 810
      CALL IFETCH('PDBED2  ',4,4,ISTAT)
      IF (ISTAT.NE.0) GO TO 820
      CALL CLOSE(4,ISTAT)
      CALL PDBED2(DBN)
      CALL CLOSE(3,ISTAT)
      GO TO 30
C     >>>>>>>  KEYS MODE  <<<<<<<
650   CALL OPENW(4,'CAL:PDBKEY.OVY  ',0,0,0,ISTAT)
      IF (ISTAT.NE.0) GO TO 810
      CALL IFETCH('PDBKEY  ',4,4,ISTAT)
      IF (ISTAT.NE.0) GO TO 820
      CALL CLOSE(4,ISTAT)
      CALL PDBKEY(DBN)
      CALL CLOSE(3,ISTAT)
      GO TO 30
C     >>>>>>>>  ROSTER MODE  <<<<<<<<
675   CALL OPENW(4,'CAL:PDBROS.OVY  ',0,0,0,ISTAT)
      IF (ISTAT.NE.0) GO TO 810
      CALL IFETCH('PDBROS  ',4,4,ISTAT)
      IF (ISTAT.NE.0) GO TO 820
      CALL CLOSE(4,ISTAT)
      CALL PDBROS(DBN)
      CALL CLOSE(3,ISTAT)
      GO TO 30
C     >>>>>>>>>>  STOP PROCESSING  <<<<<<<<<<
700   WRITE(6,1008,ERR=800)
      STOP
C     >>>>>>>>>>  ERROR REPORTING  <<<<<<<<<<
```

```
800   WRITE(6,1009,ERR=801)
801   STOP
C     >>>>>>>>>>  OVERLAY OPEN ERROR  <<<<<<<<<<
810   WRITE(6,1010,ERR=800) IMODE(1),ISTAT
      GO TO 30
C     >>>>>>>>>>  OVERLAY FETCH ERROR  <<<<<<<<<<
820   WRITE(6,1011,ERR=800) IMODE(1),ISTAT
      CALL CLOSE(4,ISTAT)
      GO TO 30
1000  FORMAT(//1X,'CBT PUBLICATIONS DATA BASE (REV 02.00-12/14/79)')
1001  FORMAT(/10X,'ENTER THE DATA BASE NAME YOU',1X,
     - 'DESIRE'/5X,'- - FAA - - REHAB - - KGS - - AUTH740 - -'/1X,
     - '- - AUTH741 - - AUTH742 - - AUTH743 - - AUTH744 - -')
1002  FORMAT(5A4)
1003  FORMAT(/1X,'ILLEGAL DATA BASE NAME..RE-ENTER')
1004  FORMAT(/1X,'ENTER DESIRED RUN MODE FOR',1X,
     - 'REHAB DATA BASE'/1X)
1005  FORMAT(/1X,'ENTER DESIRED RUN MODE FOR'/1X,
     - 'K G S   DATA BASE'/1X)
1006  FORMAT(/1X,'ENTER DESIRED RUN MODE FOR'/1X,
     - 'CBT AUTHOR DATA BASE'/1X)
1007  FORMAT(1X,'- - EDIT - - RETRIEVE - - KEYS',
     - 1X,'- - STOP - - SELECT DB - - ONE KEY - - ROSTER - -')
1008  FORMAT(1X,'* * *  END OF PROCESSING * * *')
1009  FORMAT(1X,'TERMINAL ERROR OCCURED')
1010  FORMAT(1X,'ERROR OPENING OVERLAY FOR',
     - 1X,'MODE ',A4,' STATUS=',I3)
1011  FORMAT(1X,'ERROR FETCHING OVERLAY FOR',
     - 1X,'MODE ',A4,' STATUS=',I3)
1012  FORMAT(/1X,'ILLEGAL RUN MODE...RE-ENTER')
1013  FORMAT(3A4)
1014  FORMAT(/1X,'ENTER DESIRED RUN MODE FOR'/1X,
     -  'FAA DATA BASE'/1X)
      END
.U      EXT FUNC
IDATE   INT4 VAR
DBN     INT4 VAR
IMODE   INT4 VAR
BYTE1   INT4 VAR
BYTE2   INT4 VAR
KEY     INT4 VAR
AUTDIV  INT4 VAR
ISIJH   INT4 VAR
DATE    EXT FUNC
30      LABEL
1000    LABEL
@H      EXT FUNC
70      LABEL
55      LABEL
1001    LABEL
60      LABEL
1013    LABEL
```

```
800       LABEL
I         INT4 VAR
1003      LABEL
1004      LABEL
1005      LABEL
1014      LABEL
65        LABEL
67        LABEL
68        LABEL
1006      LABEL
1007      LABEL
90        LABEL
1002      LABEL
350       LABEL
500       LABEL
530       LABEL
560       LABEL
590       LABEL
620       LABEL
650       LABEL
700       LABEL
600       LABEL
675       LABEL
ILBYTE    EXT FUNC
400       LABEL
1012      LABEL
351       LABEL
360       LABEL
411       LABEL
405       LABEL
ISHFT     EXT FUNC
ISBYTE    EXT FUNC
410       LABEL
OPENW     EXT FUNC
ISTAT     INT4 VAR
810       LABEL
IFETCH    EXT FUNC
820       LABEL
CLOSE     EXT FUNC
PDBQK     EXT FUNC
REHAEX    EXT FUNC
KGSEX     EXT FUNC
AUTHEX    EXT FUNC
FAAEX     EXT FUNC
PDBED     EXT FUNC
PDBED2    EXT FUNC
PDBKEY    EXT FUNC
PDBROS    EXT FUNC
1008      LABEL
.S        EXT FUNC
1009      LABEL
801       LABEL
```

```
1010    LABEL
1011    LABEL
.V      EXT FUNC

   0 ERRORS: FORTRAN V LEVEL 1 R04-00
```

```
$BATCH
      SUBROUTINE PDBQK(KEY,DBNAME)
C     WRITTEN BY GEORGE SMITH - CBT
C     THIS SUBROUTINE SEARCHES A PUBLICATIONS DATABASE
C FOR A PARTICULAR KEY WORD.  EACH RECORD IS CHECKED
C FOR HAVING THE KEY WORD.  THIS SUBROUTINE MAY BE INVOKED
C FROM "PDBSUP" BY TYPING 'ONE KEY', OR BY TYPING AN
C ASTERISK IN COLUMN 1 IMMEDIATELY FOLLOWED BY THE KEY
C WORD.  THE NUMBER OF RECORDS HAVING THIS KEY WORD
C IS OUTPUT TO THE TERMINAL.
C
C     MODIFIED 10/29/79 - 08:58
C     MODIFIED 10/30/79 - 08:24
      INTEGER KEY(4),RECPTR,ISTAT,IPBLK(5),I,J,INDEX1,INDEX2
      INTEGER RECS,RECORD(93),BYTE1,DBNAME(4),ORGST(25),INST(25)
      RECPTR = -1
      RECS = 0
      CALL OPENW(3,DBNAME,0,0,0,ISTAT)
      IF(ISTAT .NE. 0)GO TO 500
      DO 50 I=1,25
      ORGST(I)='    '
      INST(I)='    '
50    CONTINUE
100   RECPTR = RECPTR + 1
      CALL SYSIO(IPBLK,Y'4D',3,RECORD,372,RECPTR)
      IF(IPBLK(1) .NE. Y'4D030000')GO TO 400
      J = 62
150   J = J + 4
      IF(J .EQ. 90)GO TO 300
      DO 200 INDEX1 = 1,4
        INDEX2 = INDEX1 + J - 1
        IF(RECORD(INDEX2) .NE. KEY(INDEX1))GO TO 150
200     CONTINUE
250   RECS=RECS+1
      GO TO 100
C     ***  SEARCH TITLE FOR KEY WORD  ***
300   II=28
      DO 320 I=1,25
      ORGST(I)=RECORD(II)
      II=II+1
320   CONTINUE
      DO 350 I=1,4
      INST(I)=KEY(I)
350   CONTINUE
      CALL STSRCH(ORGST,INST,ISTAT)
      IF (ISTAT.EQ.1) GO TO 250
      GO TO 100
400   CALL ILBYTE(BYTE1,IPBLK(1),2)
      IF(BYTE1 .NE. Y'00000088')GO TO 450
405   WRITE(6,410,ERR = 405)RECS,(KEY(I),I = 1,4)
410   FORMAT('0',I4,' RECORD/S HAVE THE KEYWORD ',1H',4A4,1H'/4X
     - 'IN EITHER THE KEY WORD OR TITLE FIELD')
```

```
      GO TO 600
450   WRITE(6,460,ERR = 450)IPBLK(1)
460   FORMAT('0ERROR IN READING DATA FILE - ',Z8)
      GO TO 600
500   WRITE(6,510,ERR = 500)ISTAT
510   FORMAT('0ERROR IN OPENING DATA FILE - ',I3)
600   CALL CLOSE(3,ISTAT)
      RETURN
      END
PDBQK    FUNC/SUB
.Q       EXT FUNC
.P       EXT FUNC
KEY      FORM PAR
DBNAME   FORM PAR
RECPTR   INT4 VAR
ISTAT    INT4 VAR
IPBLK    INT4 VAR
I        INT4 VAR
J        INT4 VAR
INDEX1   INT4 VAR
INDEX2   INT4 VAR
RECS     INT4 VAR
RECORD   INT4 VAR
BYTE1    INT4 VAR
ORGST    INT4 VAR
INST     INT4 VAR
OPENW    EXT FUNC
500      LABEL
50       LABEL
100      LABEL
SYSIO    EXT FUNC
400      LABEL
150      LABEL
300      LABEL
200      LABEL
250      LABEL
II       INT4 VAR
320      LABEL
350      LABEL
STSRCH   EXT FUNC
ILBYTE   EXT FUNC
450      LABEL
405      LABEL
410      LABEL
@H       EXT FUNC
600      LABEL
460      LABEL
510      LABEL
CLOSE    EXT FUNC

   0 ERRORS: FORTRAN V LEVEL 1 R04-00
```

```
      SUBROUTINE STSRCH(ORGST,INST,ISTAT)
C     PROGRAM DESIGNED TO SEARCH AN ASCII
C     FIELD AGAINST AN INPUT FIELD TO
C     DETERMINE A MATCH CONDITION.
C     WRITTEN BY L. KAETZEL - CBT - 02/23/79
C     >>>>>>>  ARGUMENT DEFINITIONS  <<<<<<<<<
C     ORGST = ORGINAL STRING MAX 100 CHARACTERS
C     INST = INPUT STRING MAX 100 CHARACTERS
C     STAT = DISPOSITION OF PROCESSING
C          0=NO MATCH
C          1=MATCH
C         -2=NON ASCII CHARACTERS IN STRING
      INTEGER ORGST(25),INST(25)
      INTEGER OSTOR,ISTOR,OSCT,TCCT,BLNK
C     >>>>>>>  FIND NO. OF CHAR FOR SEARCH  <<<<<<<
      NCHAR=-1
      ITMP='1111'
      DO 30 I=1,25
      DO 20 J=1,4
      ISCON=32-(8*J)
      ICHAR=ISHFT(INST(I),-ISCON)
      ISTOR='    '
      CALL ISBYTE(ICHAR,ISTOR,0)
      IF (ISTOR.EQ.'    '.AND.ITMP.EQ.'    ')
     - GO TO 40
      NCHAR=NCHAR+1
      ITMP=ISTOR
20    CONTINUE
30    CONTINUE
C     >>>>>>>  VALIDATE ASCII STRINGS  <<<<<<<
40    ICOUNT=0
      NDIF=(NCHAR/4)+1
      DO 100 I=1,25
      DO 50 J=1,4
      ISCON=32-(8*J)
      ICHAR=ISHFT(ORGST(I),-ISCON)
      ISTOR='    '
      CALL ISBYTE(ICHAR,ISTOR,0)
      IF (ISTOR.LT.Y'20202020'.OR.
     - ISTOR.GT.Y'5A202020') GO TO 600
      ICHAR=ISHFT(INST(I),-ISCON)
      ISTOR='    '
      CALL ISBYTE(ICHAR,ISTOR,0)
      IF (ISTOR.LT.Y'20202020'.OR.
     - ISTOR.GT.Y'5A202020') GO TO 600
      ICOUNT=ICOUNT+1
      IF (ICOUNT.EQ.NCHAR) GO TO 150
50    CONTINUE
100   CONTINUE
C     >>>>>>>  MASK IT  <<<<<<<
150   OSCT=1
      ISCT=1
```

```
      TCCT=1
      II=1
      I=1
      ISTMP='AAAA'
200   ISCON=32-(8*I)
      IOCHAR=ISHFT(ORGST(OSCT),-ISCON)
      IOSTOR='    '
      CALL ISBYTE(IOCHAR,IOSTOR,0)
      IF (IOSTOR.EQ.'    '.AND.ISTMP.EQ.'    ') GO TO 275
      ISTMP=IOSTOR
      ISCON=32-(8*II)
      IICHAR=ISHFT(INST(ISCT),-ISCON)
      IISTOR='    '
      CALL ISBYTE(IICHAR,IISTOR,0)
      IF (IOSTOR.NE.IISTOR) GO TO 300
C     >>>>>>>  CHARACTER MATCH  <<<<<<<
      TCCT=TCCT+1
      IF (TCCT.GT.NCHAR) GO TO 500
      II=II+1
215   I=I+1
      IF (II.LE.4.AND.I.LE.4) GO TO 200
      IF (II.LE.4) GO TO 250
      II=1
      ISCT=ISCT+1
      IF (ISCT.LE.25) GO TO 250
      ISTAT=1
      RETURN
250   IF (I.LE.4) GO TO 200
      I=1
      OSCT=OSCT+1
      IF (OSCT.LE.25) GO TO 200
275   ISTAT=0
      RETURN
C     >>>>>>>  CHARACTER NO-MATCH  <<<<<<<
300   ISCT=1
      II=1
      TCCT=1
      GO TO 215
500   ISTAT=1
      RETURN
600   ISTAT=-2
      RETURN
      END
STSRCH  FUNC/SUB
.Q      EXT FUNC
.P      EXT FUNC
ORGST   FORM PAR
INST    FORM PAR
ISTAT   FORM PAR
OSTOR    INT4 VAR
ISTOR    INT4 VAR
OSCT     INT4 VAR
```

```
TCCT     INT4 VAR
BLNK     INT4 VAR
NCHAR    INT4 VAR
ITMP     INT4 VAR
30       LABEL
I        INT4 VAR
20       LABEL
J        INT4 VAR
ISCON    INT4 VAR
ICHAR    INT4 VAR
ISHFT    EXT FUNC
ISBYTE   EXT FUNC
40       LABEL
ICOUNT   INT4 VAR
NDIF     INT4 VAR
100      LABEL
50       LABEL
600      LABEL
150      LABEL
ISCT     INT4 VAR
II       INT4 VAR
ISTMP    INT4 VAR
200      LABEL
IOCHAR   INT4 VAR
IOSTOR   INT4 VAR
275      LABEL
IICHAR   INT4 VAR
IISTOR   INT4 VAR
300      LABEL
500      LABEL
215      LABEL
250      LABEL

   0 ERRORS: FORTRAN V LEVEL 1 R04-00
```

```
      SUBROUTINE PDBKEY(DBNAME)
C     MODIFIED 04/13/79 - 08:06
C     MODIFIED 10/29/79 - 13:51
C     MODIFIED 10/30/79 - 08:29
      INTEGER*4 TEMP(4,6),KEY(4,700),CHECK,ISTAT,DBNAME(4)
     $,TOTAL,IPBLK1(5),RECORD(93),I,J,T,IDATE(3),J2
      INTEGER*4 PRFILE(4)
      INTEGER*2 ANSWER,TOSORT,SWAP,END,COUNT(700)
      INTEGER*2 PAGES,BEGIN,LOOPCT,OFFSET,RECS,OCCURS
      INTEGER*2 DELETD
      EQUIVALENCE (TEMP(1,1),RECORD(66))
      RECS = 0
      DELETD = 0
      OCCURS = 0
      TOTAL = 0
      PRFILE(1) = 'PR2:'
      PRFILE(2) = '    '
      PRFILE(3) = '    '
      PRFILE(4) = '    '
19    WRITE(6,20,ERR = 19)
20    FORMAT('0BEGIN KEYWORD SORT')
      CALL CLOSE(3,ISTAT)
      CALL OPENW(3,DBNAME,0,0,0,ISTAT)
      IF(ISTAT .EQ. 0)GO TO 30
24      WRITE(6,25,ERR = 24)ISTAT
25      FORMAT('0ERROR IN OPENING DATABASE FILE '
     $          I3)
        GO TO 640
30    WRITE(6,1001,ERR=800)
      ILU=0
34    READ(6,1002,ERR=800) IOPT
      IF (IOPT.EQ.'YES ') ILU=7
      IF (IOPT.EQ.'NO  ') ILU=6
      IF (ILU.NE.0) GO TO 38
      WRITE(6,1003,ERR=800)
      GO TO 34
38    IF (ILU.EQ.6) GO TO 50
      CALL CLOSE(7,ISTAT)
      CALL OPENW(7,PRFILE,7,0,0,ISTAT)
      IF(ISTAT .EQ. 0)GO TO 50
44      WRITE(6,45,ERR = 44)ISTAT
45      FORMAT('0ERROR IN OPENING PRINT FILE')
        GO TO 640
50    CALL SYSIO(IPBLK1,Y'48',3,RECORD,372,0)
      IF(IPBLK1(1) .NE. Y'49030000')GO TO 65
      RECS = RECS + 1
      DO 55 I = 1,5
        IF(RECORD(I) .NE. 'DELE')GO TO 100
55      CONTINUE
      DELETD = DELETD + 1
      GO TO 50
65    CALL ILBYTE(CHECK,IPBLK1(1),2)
```

```
      IF(CHECK .EQ. Y'88')GO TO 158
70    WRITE(6,72,ERR = 70)IPBLK1(1)
72    FORMAT('0','ERROR:  IPBLK(1) = ',Z8)
      GO TO 640
100   DO 150 J = 1,6
        DO 120 I = 1,4
          IF(TEMP(I,J) .NE. '    ')GO TO 125
120       CONTINUE
        GO TO 150
125     IF(TOTAL .EQ. 0)GO TO 132
        DO 130 T = 1,TOTAL
          DO 127 I = 1,4
            IF(TEMP(I,J) .NE. KEY(I,T))GO TO 130
127         CONTINUE
          COUNT(T) = COUNT(T) + 1
          OCCURS = OCCURS + 1
          GO TO 150
130       CONTINUE
132     TOTAL = TOTAL + 1
        IF(TOTAL .EQ. 701)GO TO 550
        COUNT(TOTAL) = 1
        OCCURS = OCCURS + 1
        DO 135 I = 1,4
          KEY(I,TOTAL) = TEMP(I,J)
135       CONTINUE
150     CONTINUE
      GO TO 50
158   TOSORT = TOTAL
160   TOSORT = TOSORT - 1
      IF(TOSORT .EQ. 0)GO TO 280
      SWAP = 'N'
      DO 180 J = 1,TOSORT
        DO 170 I = 1,4
          IF(KEY(I,J) .GT. KEY(I,J + 1))GO TO 172
          IF(KEY(I,J) .LT. KEY(I,J + 1))GO TO 180
170       CONTINUE
        GO TO 180
172     DO 175 I = 1,4
          TEMP(1,1) = KEY(I,J)
          KEY(I,J) = KEY(I,J + 1)
          KEY(I,J + 1) = TEMP(1,1)
175       CONTINUE
        SWAP = 'Y'
        TEMP(1,1) = COUNT(J)
        COUNT(J) = COUNT(J + 1)
        COUNT(J + 1) = TEMP(1,1)
180     CONTINUE
      IF(SWAP .EQ. 'Y')GO TO 160
280   BEGIN = -45
      PAGES = TOTAL / 90
      CALL DATE(IDATE)
400   PAGES = PAGES - 1
```

```
403   WRITE(ILU,407,ERR = 403)DBNAME(2),DBNAME(3),IDATE(2)
     $                        ,IDATE(3),IDATE(1)
407   FORMAT('1',5X,2A4,2X,'PUBLICATIONS DATA BASE',5X,'KEY WORD LIST',
     $       5X,I2,'/',I2,'/',I2)
408   WRITE(ILU,409,ERR = 408)
409   FORMAT('0',5X,'KEY WORD',7X,'OCCURRENCES',18X,'KEY WORD',
     $       7X,'OCCURRENCES')
40900 WRITE(ILU,40901,ERR = 40900)
40901 FORMAT(' ',5X,'--- ----',7X,'-----------',18X,'--- ----',
     $       7X,'-----------')
      IF(PAGES .EQ. -1)GO TO 500
      LOOPCT = 0
      ICRT=0
      BEGIN = BEGIN + 45
410   LOOPCT = LOOPCT + 1
      ICRT=ICRT+1
      BEGIN = BEGIN + 1
      END = BEGIN + 45
429   WRITE(ILU,430,ERR = 429)(KEY(I,BEGIN),I = 1,4),COUNT(BEGIN),
     $                        (KEY(I,END),I = 1,4),COUNT(END)
430   FORMAT(5X,4A4,2X,I3,23X,4A4,2X,I3)
      IF (ILU.EQ.7) GO TO 450
      IF (LOOPCT.EQ.45) GO TO 400
      IF (ICRT.LT.18) GO TO 410
      READ(6,1000) IDUM
      ICRT=0
      GO TO 410
450   IF (LOOPCT.EQ.45) GO TO 400
                        GO TO 410
500   BEGIN = BEGIN + 45
      LOOPCT = 0
      OFFSET = (TOTAL - TOTAL / 90 * 90) / 2
      IF(TOTAL / 2 * 2 .NE. TOTAL)OFFSET = OFFSET + 1
505   BEGIN = BEGIN + 1
      LOOPCT = LOOPCT + 1
      IF(LOOPCT .GE. OFFSET)GO TO 550
      END = BEGIN + OFFSET
509   WRITE(ILU,430,ERR = 509)(KEY(I,BEGIN),I = 1,4),COUNT(BEGIN),
     $                        (KEY(I,END),I = 1,4),COUNT(END)
      GO TO 505
550   IF(TOTAL / 2 * 2 .NE. TOTAL)GO TO 570
      IF(OFFSET .EQ. 0)GO TO 600
      END = BEGIN + OFFSET
559   WRITE(ILU,430,ERR = 559)(KEY(I,BEGIN),I = 1,4),COUNT(BEGIN),
     $                        (KEY(I,END),I = 1,4),COUNT(END)
      GO TO 600
570   WRITE(ILU,430,ERR = 570)(KEY(I,BEGIN),I = 1,4),COUNT(BEGIN)
600   WRITE(ILU,407,ERR = 600)DBNAME(2),DBNAME(3),IDATE(2)
     $                        ,IDATE(3),IDATE(1)
610   WRITE(ILU,615,ERR = 610)RECS,DELETD,TOTAL,OCCURS
615   FORMAT('0'/'0','END OF KEY WORDS'/
     $       '0',I4,' RECORDS READ'/
```

```
     $      '0',I4,' HAD BEEN DELETED'/
     $      '0',I4,' KEY WORDS'/
     $      '0',I4,' OCCURRENCES')
620   WRITE(6,630,ERR = 620)
630   FORMAT('0END OF KEYWORD SORT')
640   CALL CLOSE(3,ISTAT)
      CALL CLOSE(7,ISTAT)
      RETURN
650   WRITE(ILU,651,ERR = 650)
651   FORMAT('0','MORE THAN 700 KEYS'
     $       /' ONLY FIRST 700 PROCESSED')
      TOTAL = TOTAL - 1
      GO TO 158
C     >>>>>>>  T E R M I N A L   E R R O R   <<<<<<<<<<
800   WRITE(6,1004)
      STOP
1000  FORMAT(A1)
1001  FORMAT(1X,'OPTIONAL PRINTER OUTPUT? (YES OR NO)')
1002  FORMAT(A4)
1003  FORMAT(1X,'RESPONSE NOT "YES" OR "NO"..RE-ENTER')
1004  FORMAT(1X,'TERMINAL ERROR OCCURED')
      END
PDBKEY  FUNC/SUB
.Q      EXT FUNC
.P      EXT FUNC
DBNAME  FORM PAR
TEMP    INT4 VAR
KEY     INT4 VAR
CHECK   INT4 VAR
ISTAT   INT4 VAR
TOTAL   INT4 VAR
IPBLK1  INT4 VAR
RECORD  INT4 VAR
I       INT4 VAR
J       INT4 VAR
T       INT4 VAR
IDATE   INT4 VAR
J2      INT4 VAR
PRFILE  INT4 VAR
ANSWER  INT2 VAR
TOSORT  INT2 VAR
SWAP    INT2 VAR
END     INT2 VAR
COUNT   INT2 VAR
PAGES   INT2 VAR
BEGIN   INT2 VAR
LOOPCT  INT2 VAR
OFFSET  INT2 VAR
RECS    INT2 VAR
OCCURS  INT2 VAR
DELETD  INT2 VAR
19      LABEL
```

```
20        LABEL
@H        EXT FUNC
CLOSE     EXT FUNC
OPENW     EXT FUNC
30        LABEL
24        LABEL
25        LABEL
640       LABEL
1001      LABEL
800       LABEL
ILU       INT4 VAR
34        LABEL
1002      LABEL
IOPT      INT4 VAR
38        LABEL
1003      LABEL
50        LABEL
44        LABEL
45        LABEL
SYSIO     EXT FUNC
65        LABEL
55        LABEL
100       LABEL
ILBYTE    EXT FUNC
158       LABEL
70        LABEL
72        LABEL
150       LABEL
120       LABEL
125       LABEL
132       LABEL
130       LABEL
127       LABEL
650       LABEL
135       LABEL
160       LABEL
280       LABEL
180       LABEL
170       LABEL
172       LABEL
175       LABEL
DATE      EXT FUNC
400       LABEL
403       LABEL
407       LABEL
408       LABEL
409       LABEL
40900     LABEL
40901     LABEL
500       LABEL
ICRT      INT4 VAR
410       LABEL
```

```
429     LABEL
430     LABEL
450     LABEL
1000    LABEL
IDUM    INT4 VAR
505     LABEL
550     LABEL
509     LABEL
570     LABEL
600     LABEL
559     LABEL
610     LABEL
615     LABEL
620     LABEL
630     LABEL
651     LABEL
1004    LABEL
.S      EXT FUNC

   0 ERRORS: FORTRAN V LEVEL 1 R04-00
```

```
$BATCH
      SUBROUTINE PDBED(DBNAME)
C     MODIFIED BY L. KAETZEL 09/14/79
C     MODIFIED 10/29/79 - 09:39
C     MODIFIED 10/31/79 - 06:51
      INTEGER MAX,MIN,RECPTR,RECORD(93),I,IPBLK1(5),PTRADJ
      INTEGER*2 DBCODE,ANSWER,LINE
      INTEGER MODE,DBNAME(4),CHECK,START,END,DIGITS,I2,ISTAT
      INTEGER TEMP,TEMP2(40),STOP,ASTRIX,BEGIN,END,BLANKS
      ASTRIX = '*   '
      BLANKS = '    '
      CALL CLOSE(3,ISTAT)
      DBCODE = 1
      IF(DBNAME(2) .EQ. 'REHA')DBCODE = 2
      CALL OPENW(3,DBNAME,7,0,0,ISTAT)
      IF(ISTAT .NE. 0)GO TO 80
                      GO TO 100
80    WRITE(6,81,ERR = 80)ISTAT
81    FORMAT(' ERROR IN OPENING FILE ',I3)
      GO TO 900
50    WRITE(6,51,ERR = 50)
51    FORMAT('0','BEGIN UPDATING OF DATABASE')
100   WRITE(6,110,ERR = 100)
110   FORMAT(' ENTER RUN MODE'/
     $ 'APPEND - DELETE - UPDATE - SEARCH - RETURN')
      READ(6,120)MODE
120   FORMAT(A4)
      IF(MODE .EQ. 'APPE')GO TO 200
      IF(MODE .EQ. 'DELE')GO TO 300
      IF(MODE .EQ. 'UPDA')GO TO 30601
      IF(MODE .EQ. 'SEAR')GO TO 500
      IF(MODE .EQ. 'RETU')GO TO 900
129     WRITE(6,130,ERR = 129)
130     FORMAT(' ILLEGAL RUN MODE')
        GO TO 100
C
C APPEND MODE
C
C FIND END OF FILE
C
200   MAX = 16384
      MIN = 0
205   RECPTR = (MAX + MIN) / 2
      READ(3,210,END = 220,REC = RECPTR)TEMP
210   FORMAT(A4)
      RECPTR = RECPTR + 1
      READ(3,210,END = 230,REC = RECPTR)TEMP
      MIN = RECPTR
      IF(RECPTR .GT. 16300)GO TO 290
      GO TO 205
220   IF(RECPTR .EQ. 0)GO TO 230
      MAX = RECPTR
```

```
      GO TO 205
C
C READ IN DATA
C
C DBCODE ONE IS KGS AND AUTHOR DATABASE
C DBCODE TWO IS REHAB DATABASE
C
C (PROGRAM DEFAULTS TO DBCODE ONE)
C
230   DO 2301 I = 1,93
2301    RECORD(I) = BLANKS
      IF(DBCODE .EQ. 1)GO TO 231
      IF(DBCODE .EQ. 2)GO TO 23410
231   WRITE(6,232,ERR = 231)
232   FORMAT(' ENTER 3 AUTHORS, ONE TO A ',
     $       'LINE'/' OR AN "*" TO EXIT')
      READ(6,234)(RECORD(I),I = 1,8)
234   FORMAT(8A4)
      IF(RECORD(1) .EQ. ASTRIX)GO TO 100
      READ(6,234)(RECORD(I),I = 9,16)
      IF(RECORD(9) .NE. ASTRIX)GO TO 23401
        RECORD(9) = BLANKS
        GO TO 249
23401 READ(6,234)(RECORD(I),I = 17,24)
      IF(RECORD(17) .NE. ASTRIX)GO TO 235
        RECORD(17) = BLANKS
        GO TO 249
23410 WRITE(6,23415,ERR = 23410)
23415 FORMAT(' ENTER NAME OF AUTHOR'/
     $       ' OR AN "*" TO EXIT')
      READ(6,234)(RECORD(I),I = 1,8)
      IF(RECORD(1) .EQ. ASTRIX)GO TO 100
23418 WRITE(6,23420,ERR = 23418)
23420 FORMAT(' ENTER LIBRARY LOCATION')
      READ(6,234)(RECORD(I),I = 9,16)
      IF(RECORD(9) .NE. ASTRIX)GO TO 23423
        RECORD(9) = BLANKS
        GO TO 249
23423 WRITE(6,23425,ERR = 23423)
23425 FORMAT(27H ENTER AUTHOR'S AFFILIATION)
      READ(6,234)(RECORD(I),I = 17,24)
      IF(RECORD(17) .NE. ASTRIX)GO TO 235
        RECORD(17) = BLANKS
        GO TO 249
235   WRITE(6,236,ERR = 235)
236   FORMAT(' ENTER PUBLICATION DATE')
      READ(6,234)(RECORD(I),I = 25,27)
      IF(RECORD(25) .NE. ASTRIX)GO TO 239
        RECORD(25) = BLANKS
        GO TO 249
239   WRITE(6,240,ERR = 239)
240   FORMAT(' ENTER TWO LINES OF THE TITLE')
```

```
      READ(6,241)(TEMP2(I),I = 1,20)
      IF(TEMP2(1) .EQ. ASTRIX)GO TO 249
      READ(6,241)(TEMP2(I),I = 21,40)
      STOP = TEMP2(21)
      IF(TEMP2(21) .EQ. ASTRIX)TEMP2(21) = BLANKS
241   FORMAT(20A4)
2410  CALL CRAM(TEMP2)
      DO 2411 I = 1,25
2411    RECORD(I + 27) = TEMP2(I)
      IF(STOP .EQ. ASTRIX)GO TO 249
242   WRITE(6,243,ERR = 242)
243   FORMAT(' ENTER PUBLICATION NAME')
      READ(6,234)(RECORD(I),I = 53,60)
      IF(RECORD(53) .NE. ASTRIX)GO TO 244
        RECORD(53) = BLANKS
        GO TO 249
244   WRITE(6,245,ERR = 244)
245   FORMAT(' ENTER VOLUME, NUMBER, PAGES, ETC.')
      READ(6,234)(RECORD(I),I = 61,65)
      IF(RECORD(61) .NE. ASTRIX)GO TO 24504
        RECORD(61) = BLANKS
        GO TO 249
C     >>>>>>>>>>  ENTER DATA FOR AUTHOR DATA BASE <<<<<<<<<<
24504 IF (DBNAME(2).NE.'AUTH') GO TO 246
      WRITE(6,24500)
      BEGIN=56
      END=69
      READ(6,234) (RECORD(I),I=BEGIN,END)
      WRITE(6,24501)
      BEGIN=BEGIN+4
      END=END+4
      READ(6,234) (RECORD(I),I=BEGIN,END)
      WRITE(6,24502)
      BEGIN=BEGIN+4
      END=END+4
      READ(6,234) (RECORD(I),I=BEGIN,END)
      DO 24503 I=78,93
      RECORD(I)='    '
24503 CONTINUE
      GO TO 250
24500 FORMAT(1X,'ENTER PUBLISHER LOCATION')
24501 FORMAT(1X,'ENTER PUBLISHERS NAME')
24502 FORMAT(1X,'ENTER DIVISION/GROUP')
246   WRITE(6,247,ERR = 246)
247   FORMAT(' ENTER 6 KEY WORDS, ONE TO A LINE')
      DO 248 BEGIN = 66,89,4
        END = BEGIN + 3
        READ(6,234)(RECORD(I),I = BEGIN,END)
        IF(RECORD(BEGIN) .NE. ASTRIX)GO TO 248
          RECORD(BEGIN) = BLANKS
          GO TO 249
248     CONTINUE
```

```
      IF(DBCODE .EQ. 2)GO TO 249
24800 WRITE(6,24801,ERR = 24800)
24801 FORMAT(' ENTER DISCIPLINE')
      READ(6,234)(RECORD(I),I = 90,93)
      IF(RECORD(90) .NE. ASTRIX)GO TO 249
        RECORD(90) = '    '
249   CALL DISPLY(RECORD,DBCODE)
250   WRITE(6,251,ERR = 250)
251   FORMAT(' SAVE OK?')
      READ(6,260)ANSWER
260   FORMAT(A4)
      IF(ANSWER .EQ. 'YES ')GO TO 2700
      IF(ANSWER .EQ. 'NO  ')GO TO 230
261   WRITE(6,262,ERR = 261)
262   FORMAT(' ANSWER "YES" OR "NO"')
      GO TO 250
C
C SAVE APPENDED RECORD
C
2700  CALL SYSIO(IPBLK1,Y'28',3,RECORD,372,0)
      IF(IPBLK1(1) .NE. Y'28030000')GO TO 270
      GO TO 230
270   WRITE(6,272,ERR = 270)IPBLK1(1)
272   FORMAT('0','ERROR IN WRITING RECORD'/
     $       ' IPBLK(1) = ',Z8)
      GO TO 100
290   WRITE(6,292,ERR = 290)
292   FORMAT('0','TOO MANY RECORDS ON FILE')
      GO TO 100
C
C DELETE AND UPDATE MODES
C
300   WRITE(6,305,ERR = 300)
305   FORMAT(' ENTER NUMBER OF THE RECORD TO'
     $       ,' DELETE'/' OR AN "*" TO EXIT')
      GO TO 307
30601 WRITE(6,306,ERR = 30601)
306   FORMAT(' ENTER NUMBER OF RECORD TO UPDATE'/
     $       ' OR AN "*" TO EXIT')
C
C READ NUMBER IN ALPHANUMERIC FORMAT
C OF RECORD TO DELETE OR UPDATE AND
C CONVERT NUMBER TO INTEGER
C
307   READ(6,308)DIGITS
308   FORMAT(A4)
      IF(DIGITS .EQ. ASTRIX)GO TO 100
      RECPTR = 0
      DO 310 I = 1,4
        I2 = I - 1
        CALL ILBYTE(CHECK,DIGITS,I2)
        IF(CHECK .EQ. Y'00000020')GO TO 310
```

```
        IF(CHECK .LT. Y'00000030' .OR.
     $     CHECK .GT. Y'00000039')GO TO 315
        RECPTR = RECPTR * 10 + (CHECK - 48)
310     CONTINUE
      GO TO 320
315   WRITE(6,317,ERR = 315)
317   FORMAT('0','THAT IS NOT AN ASTERISK'/
     $       22H AND IT'S NOT A NUMBER)
      IF(MODE .EQ. 'UPDA')GO TO 30601
      GO TO 300
C
C READ RECORD TO DELETE OR UPDATE
C
320   PTRADJ = RECPTR - 1
      CALL SYSIO(IPBLK1,Y'4C',3,RECORD,372,PTRADJ)
      IF(IPBLK1(1) .NE. Y'4C030000')GO TO 360
      DO 321 I = 1,93
        IF(RECORD(I) .NE. 'DELE')GO TO 323
321     CONTINUE
322   WRITE(6,3220,ERR = 322)RECPTR
3220  FORMAT('0RECORD NUMBER ',I4/
     $       ' HAS ALREADY BEEN DELETED')
      IF(MODE .EQ. 'DELE')GO TO 300
                          GO TO 30601
323   WRITE(6,325,ERR = 323)RECPTR,(RECORD(I),I
     $        = 1,8)
325   FORMAT(' RECORD NUMBER ',I4/
     $       ' AUTHOR: ',8A4)
326   WRITE(6,327,ERR = 326)(RECORD(I),I = 28,52)
327   FORMAT(' TITLE: ',12A4/' ',13A4)
      IF(MODE .EQ. 'UPDA')GO TO 390
328   WRITE(6,329,ERR = 328)
329   FORMAT(' DELETE OK?')
      READ(6,330)ANSWER
330   FORMAT(A4)
      IF(ANSWER .EQ. 'NO  ')GO  TO 300
      IF(ANSWER .EQ. 'YES ')GO TO 340
332     WRITE(6,333,ERR = 332)
333   FORMAT(' ANSWER "YES" OR "NO"')
        GO TO 328
C
C DELETE RECORD
C
340   DO 345 I = 1,93
345     RECORD(I) = 'DELE'
      CALL SYSIO(IPBLK1,Y'2C',3,RECORD,372,PTRADJ)
      IF(IPBLK1(1) .NE. Y'2C030000')GO TO 370
349   WRITE(6,350,ERR = 349)
350   FORMAT(' RECORD DELETED')
      GO TO 300
360   CALL ILBYTE(CHECK,IPBLK1(1),2)
      IF(CHECK .EQ. Y'88')GO TO 365
```

```
                        GO TO 370
365   WRITE(6,367,ERR = 365)RECPTR
367   FORMAT(' RECORD NUMBER ',I4,' IS PAST THE END OF THE FILE')
      IF(MODE .EQ. 'UPDA')GO TO 30601
      GO TO 300
370   WRITE(6,373,ERR = 370)IPBLK1(1)
373   FORMAT(' ERROR'/' IPBLK(1) = ',Z8)
      GO TO 100
C
C DISPLAY RECORD TO UPDATE
C
390   CALL DISPLY(RECORD,DBCODE)
C
C GET NUMBER OF LINE TO UPDATE
C
440   WRITE(6,445,ERR = 440)
445   FORMAT(' ENTER LINE NUMBER TO UPDATE'/
     $       ' OR AN "*" TO EXIT')
      READ(6,450)DIGITS
450   FORMAT(A2)
      IF(DIGITS .EQ. '*')GO TO 490
      LINE = 0
      DO 455 I = 1,2
        I2 = I - 1
        CALL ILBYTE(CHECK,DIGITS,I2)
        IF(CHECK .EQ. Y'00000020')GO TO 455
        IF(CHECK .GT. Y'00000039' .OR.
     $     CHECK .LT. Y'00000030')GO TO 456
        LINE = LINE * 10 + CHECK - 48
455     CONTINUE
      GO TO 461
456   WRITE(6,317,ERR = 456)
      GO TO 440
C
C GET NEW LINE
C
461   IF(LINE .LT. 1 .OR. LINE .GT. 15)GO TO 495
      IF(LINE .EQ. 15 .AND. DBCODE .EQ. 2)GO TO 495
      GO TO (4610,4610,4610,4611,4612,4971,4613,4614,4615,4615,4615,
     $       4615,4615,4615,4616),LINE
4610  START = (LINE - 1) * 8 + 1
      END = START + 7
      GO TO 470
4611  START = 25
      END = 27
      GO TO 470
4612  WRITE(6,46120,ERR = 4612)LINE,(RECORD(I),I = 28,52)
46120 FORMAT('0','LINE ',I2,' WAS:'/' ',13A4/' '12A4/
     $       ' ENTER 2 LINES OF THE NEW TITLE')
      READ(6,46121)(TEMP2(I),I = 1,20)
46121 FORMAT(20A4)
      READ(6,46121)(TEMP2(I),I = 21,40)
```

```
      CALL CRAM(TEMP2)
      DO 46123 I = 1,25
        RECORD(I + 27) = TEMP2(I)
46123   CONTINUE
      GO TO 440
4613  START = 53
      END = 60
      GO TO 470
4614  START = 61
      END = 65
      GO TO 470
4615  START = 4 * LINE + 30
      END = START + 3
      GO TO 470
4616  START = 90
      END = 93
470   WRITE(6,475,ERR = 470)LINE,(RECORD(I),I = START,END)
475   FORMAT('0','LINE ',I2,' WAS:'/' ',8A4/' ENTER NEW LINE')
      READ(6,480)(RECORD(I),I = START,END)
480   FORMAT(20A4)
      GO TO 440
C
C DISPLAY UPDATED RECORD
C
490   CALL DISPLY(RECORD,DBCODE)
491   WRITE(6,251,ERR = 491)
      READ(6,492)ANSWER
492   FORMAT(A4)
      IF(ANSWER .EQ. 'YES ')GO TO 493
      IF(ANSWER .EQ. 'NO  ')GO TO 30601
49201 WRITE(6,262,ERR = 262)
      GO TO 491
C
C SAVE UPDATED RECORD
C
493   CALL SYSIO(IPBLK1,Y'2C',3,RECORD,372,PTRADJ)
      IF(IPBLK1(1) .NE. Y'2C030000')GO TO 370
      GO TO 30601
495   WRITE(6,497,ERR = 495)
497   FORMAT(' THAT IS NOT A VALID LINE NUMBER')
      GO TO 440
4971  WRITE(6,4972,ERR = 4971)
4972  FORMAT(' YOU CAN ONLY EDIT LINE 6'
     $       /' AFTER YOU HAVE EDITED LINE 5')
      GO TO 440
500   CALL SEARCH
      GO TO 100
900   CALL CLOSE(3,ISTAT)
      RETURN
      END
PDBED   FUNC/SUB
.Q      EXT FUNC
```

```
.P        EXT FUNC
DBNAME    FORM PAR
MAX       INT4 VAR
MIN       INT4 VAR
RECPTR    INT4 VAR
RECORD    INT4 VAR
I         INT4 VAR
IPBLK1    INT4 VAR
PTRADJ    INT4 VAR
DBCODE    INT2 VAR
ANSWER    INT2 VAR
LINE      INT2 VAR
MODE      INT4 VAR
CHECK     INT4 VAR
START     INT4 VAR
END       INT4 VAR
DIGITS    INT4 VAR
I2        INT4 VAR
ISTAT     INT4 VAR
TEMP      INT4 VAR
TEMP2     INT4 VAR
STOP      INT4 VAR
ASTRIX    INT4 VAR
BEGIN     INT4 VAR
BLANKS    INT4 VAR
CLOSE     EXT FUNC
OPENW     EXT FUNC
80        LABEL
100       LABEL
81        LABEL
@H        EXT FUNC
900       LABEL
50        LABEL
51        LABEL
110       LABEL
120       LABEL
200       LABEL
300       LABEL
30501     LABEL
500       LABEL
129       LABEL
130       LABEL
205       LABEL
210       LABEL
220       LABEL
230       LABEL
290       LABEL
2301      LABEL
231       LABEL
23410     LABEL
232       LABEL
234       LABEL
```

```
23401   LABEL
249     LABEL
235     LABEL
23415   LABEL
23418   LABEL
23420   LABEL
23423   LABEL
23425   LABEL
236     LABEL
239     LABEL
240     LABEL
241     LABEL
2410    LABEL
CRAM    EXT FUNC
2411    LABEL
242     LABEL
243     LABEL
244     LABEL
245     LABEL
24504   LABEL
246     LABEL
24500   LABEL
24501   LABEL
24502   LABEL
24503   LABEL
250     LABEL
247     LABEL
248     LABEL
24600   LABEL
24801   LABEL
DISPLY  EXT FUNC
251     LABEL
260     LABEL
2700    LABEL
261     LABEL
262     LABEL
SYSIO   EXT FUNC
270     LABEL
272     LABEL
292     LABEL
305     LABEL
307     LABEL
306     LABEL
308     LABEL
310     LABEL
ILBYTE  EXT FUNC
315     LABEL
320     LABEL
317     LABEL
360     LABEL
321     LABEL
323     LABEL
```

```
322      LABEL
3220     LABEL
325      LABEL
326      LABEL
327      LABEL
390      LABEL
328      LABEL
329      LABEL
330      LABEL
340      LABEL
332      LABEL
333      LABEL
345      LABEL
370      LABEL
349      LABEL
350      LABEL
365      LABEL
367      LABEL
373      LABEL
440      LABEL
445      LABEL
450      LABEL
490      LABEL
455      LABEL
456      LABEL
461      LABEL
495      LABEL
4610     LABEL
4611     LABEL
4612     LABEL
4971     LABEL
4613     LABEL
4614     LABEL
4615     LABEL
4616     LABEL
470      LABEL
46120    LABEL
46121    LABEL
46123    LABEL
475      LABEL
480      LABEL
491      LABEL
492      LABEL
493      LABEL
49201    LABEL
497      LABEL
4972     LABEL
SEARCH   EXT FUNC

   0 ERRORS: FORTRAN V LEVEL 1 R04-00
```

```
      SUBROUTINE CRAM(RECORD)
C
C THIS SUBROUTINE COMPRESSES CONSECUTIVE
C BLANKS INTO ONE BLANK
C
      INTEGER TPOINT,LPOINT,CHAR,RECORD(40)
      TPOINT = -1
      LPOINT = -1
100   LPOINT = LPOINT + 1
      IF(LPOINT .EQ. 160)GO TO 300
      TPOINT = TPOINT + 1
      IF(TPOINT .EQ. 100)GO TO 400
      CALL ILBYTE(CHAR,RECORD,LPOINT)
110   CALL ISBYTE(CHAR,RECORD,TPOINT)
      IF(CHAR .EQ. Y'00000020')GO TO 120
                  GO TO 100
120   LPOINT = LPOINT + 1
      IF(LPOINT .EQ. 160)GO TO 300
      CALL ILBYTE(CHAR,RECORD,LPOINT)
      IF(CHAR .EQ. Y'00000020')GO TO 120
      IF(TPOINT .EQ. 100)GO TO 110
        TPOINT = TPOINT + 1
        IF(TPOINT .EQ. 100)GO TO 400
      GO TO 110
300   TPOINT = TPOINT + 1
      IF(TPOINT .EQ. 100)GO TO 400
        CALL ISBYTE(Y'0020',RECORD,TPOINT)
        GO TO 300
400   RETURN
      END
CRAM    FUNC/SUB
.Q      EXT FUNC
.P      EXT FUNC
RECORD  FORM PAR
TPOINT  INT4 VAR
LPOINT  INT4 VAR
CHAR    INT4 VAR
100     LABEL
300     LABEL
400     LABEL
ILBYTE  EXT FUNC
110     LABEL
ISBYTE  EXT FUNC
120     LABEL

   0 ERRORS: FORTRAN V LEVEL 1 R04-00
```

```
      SUBROUTINE SEARCH
C
C THIS SUBROUTINE SEARCHES FOR THE (FIRST)
C AUTHOR'S NAME OF EACH RECORD FOR A MATCH
C WITH AN AUTHOR'S NAME ENTERED BY THE USER.
C
C THE SUBROUTINE IS USEFUL IN FINDING THE
C RECORD NUMBER OF A RECORD
C
      INTEGER AUTHOR(8),RECPTR,IPBLK1(1),RECORD(93),I,CHECK
      INTEGER PTRADJ
      INTEGER*2 ANSWER
100   WRITE(6,110,ERR = 100)
110   FORMAT(20H ENTER AUTHOR'S NAME)
      READ(6,120)AUTHOR
120   FORMAT(8A4)
      PTRADJ = -1
125   PTRADJ = PTRADJ + 1
      CALL SYSIO(IPBLK1,Y'4C',3,RECORD,372,PTRADJ)
      IF(IPBLK1(1) .NE. Y'4C030000')GO TO 160
      DO 130 I = 1,8
        IF(RECORD(I) .NE. AUTHOR(I))GO TO 125
130     CONTINUE
      RECPTR = PTRADJ + 1
133   WRITE(6,135,ERR = 133)RECPTR,(RECORD(I),I = 1,8)
135   FORMAT(' RECORD NUMBER ',I4/' AUTHOR: ',8A4)
136   WRITE(6,138,ERR = 136)(RECORD(I),I = 28,52)
138   FORMAT(' TITLE: ',12A4/' ',13A4)
139   WRITE(6,140,ERR = 139)
140   FORMAT('0','SHALL I SEARCH FOR MORE UNDER THIS AUTHOR?')
      READ(6,145)ANSWER
145   FORMAT(A4)
      IF(ANSWER .EQ. 'YES ')GO TO 125
      IF(ANSWER .EQ. 'NO  ')RETURN
149     WRITE(6,150,ERR = 149)
150     FORMAT(' ANSWER "YES" OR "NO"')
        GO TO 139
160   CALL ILBYTE(CHECK,IPBLK1(1),2)
      IF(CHECK .EQ. Y'88')GO TO 170
                          GO TO 180
170   WRITE(6,171,ERR = 170)
171   FORMAT('0','END OF SEARCH')
      RETURN
180   WRITE(6,181,ERR = 180)IPBLK1(1)
181   FORMAT(' ERROR'/' IPBLK(1) = ',Z8)
      RETURN
      END
SEARCH  FUNC/SUB
.Q      EXT FUNC
.P      EXT FUNC
AUTHOR  INT4 VAR
RECPTR  INT4 VAR
```

```
IPBLK1   INT4 VAR
RECORD   INT4 VAR
I        INT4 VAR
CHECK    INT4 VAR
PTRADJ   INT4 VAR
ANSWER   INT2 VAR
100      LABEL
110      LABEL
@H       EXT FUNC
120      LABEL
125      LABEL
SYSIO    EXT FUNC
160      LABEL
130      LABEL
133      LABEL
135      LABEL
136      LABEL
138      LABEL
139      LABEL
140      LABEL
145      LABEL
149      LABEL
150      LABEL
ILBYTE   EXT FUNC
170      LABEL
180      LABEL
171      LABEL
181      LABEL

   0 ERRORS: FORTRAN V LEVEL 1 R04-00
```

```
      SUBROUTINE DISPLY(RECORD,DBCODE)
C
C THIS SUBROUTINE DISPLAYS A RECORD IN
C A 14 LINE FORMAT FOR VIEWING FROM A
C TERMINAL
C
      INTEGER RECORD(93)
      INTEGER*2 DBCODE
100   WRITE(6,110,ERR = 100)(RECORD(I),I = 1,24)
110   FORMAT('0'/' 1  ',8A4/' 2  ',8A4/' 3  ',8A4)
120   WRITE(6,130,ERR = 120)(RECORD(I),I = 25,52)
130   FORMAT(' 4  ',3A4/' 5  ',13A4/' 6  ',12A4)
140   WRITE(6,150,ERR = 140)(RECORD(I),I = 53,65)
150   FORMAT(' 7  ',8A4/' 8  ',5A4)
160   WRITE(6,170,ERR = 160)(RECORD(I),I = 66,89)
170   FORMAT(' 9  ',4A4/' 10 ',4A4/' 11 ',4A4/' 12 ',4A4/
     $       ' 13 ',4A4/' 14 ',4A4)
      IF(DBCODE .EQ. 2)GO TO 900
180   WRITE(6,190,ERR = 180)(RECORD(I),I = 90,93)
190   FORMAT(' 15 ',4A4)
900   RETURN
      END
DISPLY  FUNC/SUB
.Q      EXT FUNC
.P      EXT FUNC
RECORD  FORM PAR
DBCODE  FORM PAR
100     LABEL
110     LABEL
@H      EXT FUNC
I       INT4 VAR
120     LABEL
130     LABEL
140     LABEL
150     LABEL
160     LABEL
170     LABEL
900     LABEL
180     LABEL
190     LABEL

   0 ERRORS: FORTRAN V LEVEL 1 R04-00
```

```
$BATCH
      SUBROUTINE KGSEX
C     DESIGNED TO READ A CBT PUBLICATIONS DATABASE
C     AND EXTRACT RECORDS BASED ON PARAMETERS INPUT
C     FROM USER COMMAND DEVICE---INPUTS MAY BE AUTHOR,
C     OR 1 TO 6 KEYWORDS. SEARCH IS BY FIELD OR STRING
C     THIS PROGRAM IS OVERLAYED FROM PDBSUP
C     WRITTEN BY LARRY KAETZEL - CBT - 02/21/79
C     >>>>>>>>>>  LOGICAL UNIT ASSIGNMENTS  <<<<<<<<<<
C     LU6 = COMMAND I/O DEVICE
C     LU3 = DATABASE FILE
C     LU7 = OPTIONAL PRINT FILE
C     MODIFIED 06/29/80 - 14:29
C     MODIFIED 06/30/80 - 12:51
C     MODIFIED 07/01/80 - 08:30
C     MODIFIED 07/10/80 - 16:56
C     MODIFIED 07/31/80 - 15:00
C     MODIFIED 08/01/80 - 10:26
C
C     >>>>>>>>>>  RECORD FORMAT  <<<<<<<<<<
C     AUTHOR 1 - 32 CHAR - POSITION 1-8
C     AUTHOR 2 - 32 CHAR - POSITION 9-16
C     AUTHOR 3 - 32 CHAR - POSITION 17-24
C     PUBL DATE - 12 CHAR - POSITION 25-27
C     TITLE - 100 CHAR - POSITION 28-52
C     PUBL NAME - 32 CHAR - POSITION 53-60
C     REFERENCE - 20 CHAR - POSITION 61-65
C     KEY WORD 1 - 16 CHAR - POSITION 66-69
C     KEY WORD 2 - 16 CHAR - POSITION 70-73
C     KEY WORD 3 - 16 CHAR - POSITION 74-77
C     KEY WORD 4 - 16 CHAR - POSITION 78-81
C     KEY WORD 5 - 16 CHAR - POSITION 82-85
C     KEY WORD 6 - 16 CHAR - POSITION 86-89
C     DISCIPLINE - 16 CHAR - POSITION 90-93
C     >>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>>
C
      INTEGER IPBLK(5),RECORD(93),INAUTH(8),INKEY(6,4),
     - IDATE(3),ITIME(3),DBNAME(4),IKMAP(6),
     - SKEY(25),ORGST(25)
      INTEGER*2 JLU,INDEX2
      DATA DBNAME/'CAL:KGS
      IPAGE=0
      CALL DATE(IDATE)
      CALL TIME(ITIME)
      CALL OPENW(3,DBNAME,0,0,0,ISTAT)
      IF (ISTAT.NE.0) GO TO 8010
C     >>>>>>>  GET INPUT CRITERIA  <<<<<<<
100   WRITE(6,9000,ERR=8000) IDATE(2),IDATE(3),IDATE(1),
     - ITIME
      WRITE(6,9001,ERR=8000)
120   READ(6,9002,ERR=8000) IPRT
      IF (IPRT.EQ.'YES ') GO TO 140
```

```
      IF (IPRT.EQ.'NO  ') GO TO 200
      WRITE(6,9005,ERR=8000)
      GO TO 120
140   CONTINUE
160   CALL OPENW(7,'PR2:            ',4,0,0,ISTAT)
      IF (ISTAT.NE.0) GO TO 8005
200   DO 205 I=1,8
      INAUTH(I)='    '
205   CONTINUE
      DO 215 I=1,6
      DO 210 J=1,4
      INKEY(I,J)='    '
210   CONTINUE
215   CONTINUE
      DO 212 I=1,25
      ORGST(I)='    '
212   CONTINUE
      WRITE(6,9007,ERR=8000)
220   READ(6,9002,ERR=8000) IPAR
      IF (IPAR.EQ.'AUTH') GO TO 280
      IF (IPAR.EQ.'KEY ') GO TO 250
      WRITE(6,9005,ERR=8000)
      GO TO 200
C     ******* G E T  K E Y W O R D  P A R A M E T E R S  *******
250   WRITE(6,9010,ERR=8000)
      ISTYP='UNI '
      DO 260 I=1,6
      READ(6,9011,ERR=8000) (INKEY(I,J),J=1,4)
      IF (INKEY(I,1).EQ.Y'20202020')  GO TO 265
260   CONTINUE
265   IF (INKEY(2,1).EQ.Y'20202020')  GO TO 300
      WRITE(6,9035,ERR=8000)
270   READ(6,9036,ERR=8000) ISTYP
      IF (ISTYP.EQ.'UNI '.OR.ISTYP.EQ.'INT ') GO TO 300
      WRITE(6,9005,ERR=8000)
      GO TO 270
C     *******  G E T  A U T H O R  P A R A M E T E R S  *******
280   WRITE(6,9008,ERR=8000)
      READ(6,9009,ERR=8000) INAUTH
      WRITE(6,9031,ERR=8004)
285   READ(6,9002,ERR=8004) ISRC
      IF (ISRC.EQ.'STRI'.OR.ISRC.EQ.'FIEL') GO TO 300
      WRITE(6,9032,ERR=8004)
      GO TO 285
C     >>>>>>>  START DATABASE SEARCH  <<<<<<<
300   IREC=0
      IEXT=0
      ICRT=1
      IF (IPRT.EQ.'YES ') ILU=7
      IF (IPRT.EQ.'NO  ') ILU=6
310   IPAGE=IPAGE+1
      IF (ILU.EQ.7) WRITE(ILU,9028,ERR=8004) IDATE(2),IDATE(3),IDATE(1)
```

```
      WRITE(ILU,9029,ERR=8004)
      IF (INAUTH(1).NE.'    ')
     -    WRITE(ILU,9022,ERR=8004) INAUTH,IPAGE
      IF (INKEY(1,1).NE.'    ')
     - WRITE(ILU,9023,ERR=8004) (INKEY(1,J1),J1=1,4),
     - (INKEY(2,J2),J2=1,4),(INKEY(3,J3),J3=1,4),IPAGE
      IF (INKEY(1,1).NE.'    ')
     - WRITE(ILU,9024,ERR=8004) (INKEY(4,J4),J4=1,4),
     - (INKEY(5,J5),J5=1,4),(INKEY(6,J6),J6=1,4)
      IF (INKEY(2,1).NE.Y'20202020')  WRITE(ILU,9037) ISTYP
      WRITE(ILU,9029,ERR=8004)
      IRNUM=0
340   CALL SYSIO(IPBLK,Y'4D',3,RECORD,372,IREC)
      IF (IPBLK(1).NE.Y'4D030000') GO TO 500
      IREC=IREC+1
      IOR=IREC
      IF (RECORD(1).EQ.'DELE'.AND.RECORD(25).EQ.'DELE')
     - GO TO 340
      IF (INAUTH(1).EQ.'ALL ') GO TO 370
      IF (INAUTH(1).EQ.'    ') GO TO 400
C     >>>>>>>   AUTHOR SEARCH  <<<<<<<
      IF (ISRC.EQ.'FIEL') GO TO 350
      DO 342 I=1,8
      ORGST(I)=RECORD(I)
342   CONTINUE
      CALL STSRCH(ORGST,INAUTH,ISTAT)
      IF (ISTAT.EQ.1) GO TO 370
      IF (ISTAT.EQ.-2) GO TO 520
      II=9
      DO 344 I=1,8
      ORGST(I)=RECORD(II)
      II=II+1
344   CONTINUE
      CALL STSRCH(ORGST,INAUTH,ISTAT)
      IF (ISTAT.EQ.1) GO TO 370
      IF (ISTAT.EQ.-2) GO TO 520
      II=17
      DO 346 I=1,8
      ORGST(I)=RECORD(II)
      II=II+1
346   CONTINUE
      CALL STSRCH(ORGST,INAUTH,ISTAT)
      IF (ISTAT.EQ.1) GO TO 370
      IF (ISTAT.EQ.-2) GO TO 520
      GO TO 340
350   DO 360 I=1,8
      IF (INAUTH(I).NE.RECORD(I)) GO TO 340
360   CONTINUE
C     >>>>>>>   CRITERIA FOUND - WRITE IT OUT  <<<<<<<
370   WRITE(ILU,9014,ERR=8004) (RECORD(I),I=1,8),IOR
      WRITE(ILU,9015,ERR=8004) (RECORD(I),I=9,16)
      WRITE(ILU,9016,ERR=8004) (RECORD(I),I=17,24)
```

```
      WRITE(ILU,9030,ERR=8004) (RECORD(K1),K1=90,93)
      WRITE(ILU,9017,ERR=8004) (RECORD(I),I=25,27)
      WRITE(ILU,9018,ERR=8004) (RECORD(I),I=28,40)
      WRITE(ILU,90181,ERR=8004) (RECORD(I),I=41,52)
      WRITE(ILU,9019,ERR=8004) (RECORD(I),I=53,60),
     - (RECORD(II),II=61,65)
      WRITE(ILU,9020,ERR=8004) (RECORD(K1),K1=66,69),
     - (RECORD(K2),K2=70,73),(RECORD(K3),K3=74,77)
      WRITE(ILU,9021,ERR=8004) (RECORD(K4),K4=78,81),
     - (RECORD(K5),K5=82,85),(RECORD(K6),K6=86,89)
      WRITE(ILU,9029,ERR=8004)
      IF (ILU.EQ.7) GO TO 372
      ICRT=ICRT+1
      IF (ICRT.LT.3) GO TO 372
      READ(6,9034,ERR=8004) IIDUM
      ICRT=1
372   IEXT=IEXT+1
      IRNUM=IRNUM+1
      IF (IRNUM.GT.3) GO TO 310
      GO TO 340
C     >>>>>>>   KEY WORD SEARCH   <<<<<<<
400   DO 405 I=1,6
      IKMAP(I)=0
405   CONTINUE
      DO 410 I=1,25
      ORGST(I)=Y'20202020'
      SKEY(I)=Y'20202020'
410   CONTINUE
420   IKEY=1
430   DO 435 I=1,4
      SKEY(I)=INKEY(IKEY,I)
435   CONTINUE
      IKST=66
      IKSP=69
438   J=1
      DO 440 I=IKST,IKSP
      ORGST(J)=RECORD(I)
      J=J+1
440   CONTINUE
      CALL STSRCH(ORGST,SKEY,ISTAT)
      IF (ISTAT.EQ.1) IKMAP(IKEY)=1
445   IKST=IKST+4
      IKSP=IKSP+4
      IF (RECORD(IKST).EQ.Y'20202020') GO TO 450
      IF (IKSP.GT.89) GO TO 450
      GO TO 438
450   IKEY=IKEY+1
      IF (IKEY.GT.6) GO TO 462
      IF (INKEY(IKEY,1).EQ.Y'20202020') GO TO 462
      GO TO 430
C     *******   S E T U P   K E Y W O R D   T I T L E   S E A R C H  ******
462   JJ=28
```

```
      DO 463 KK=1,25
      SKEY(KK)='    '
      ORGST(KK)=RECORD(JJ)
      JJ=JJ+1
463   CONTINUE
      J=1
465   DO 470 I=1,4
      IF (J.GT.6) GO TO 480
      IF (INKEY(J,1).EQ.Y'20202020'.AND.ISTYP.EQ.'UNI ')
     -   GO TO 475
      IF (INKEY(J,1).EQ.Y'20202020'.AND.ISTYP.EQ.'INT ')
     -  GO TO 480
      SKEY(I)=INKEY(J,I)
470   CONTINUE
      CALL STSRCH(ORGST,SKEY,ISTAT)
      IF (ISTAT.EQ.1.AND.ISTYP.EQ.'INT ') IKMAP(J)=1
      IF (ISTAT.EQ.1.AND.ISTYP.EQ.'UNI ') GO TO 370
      IF (ISTAT.EQ.-2) GO TO 520
      J=J+1
      GO TO 465
C     ******* C H E C K  H I T - " O R "  C O N D I T I O N *******
475   DO 478 I=1,6
      IF (IKMAP(I).EQ.1)  GO TO 370
478   CONTINUE
      GO TO 340
C     ******* C H E C K  H I T  -  " A N D "  C O N D I T I O N  *******
480   IKCNT=0
      IHIT=0
      DO 495 I=1,6
      IF (INKEY(I,1).NE.Y'20202020') IKCNT=IKCNT+1
495   CONTINUE
      IF (IKCNT.EQ.0) GO TO 340
      DO 498 I=1,6
      IF (IKMAP(I).NE.0) IHIT=IHIT+1
498   CONTINUE
      IF (IHIT.EQ.IKCNT) GO TO 370
      GO TO 340
500   IF (IPBLK(1).NE.Y'4D038880') GO TO 8015
      JLU = 5
50000 JLU = JLU + 1
      IF(JLU .EQ. 8)GO TO 50008
        IF(ISRC .EQ. 'STRI')GO TO 50001
                            GO TO 50002
50001   WRITE(JLU,90261,ERR = 8000)
90261   FORMAT('0END OF STRING SEARCH ON')
        GO TO 50003
50002   WRITE(JLU,90262,ERR = 8000)
90262   FORMAT('0END OF FIELD SEARCH ON')
50003   IF(IPAR .EQ. 'AUTH')GO TO 50004
                            GO TO 50005
50004   WRITE(JLU,90263,ERR = 8000)INAUTH
90263   FORMAT(' AUTHOR ',8A4)
```

```
      GO TO 50007
50005 WRITE(JLU,90264,ERR = 8000)((INKEY(I,INDEX2)
     $                    ,INDEX2 = 1,4),I = 1,6)
90264 FORMAT(' KEYWORDS:'/4X,4A4,3X,4A4,3X,4A4/
     $       4X,4A4,3X,4A4,3X,4A4)
50007 WRITE(JLU,90265,ERR = 50007)IEXT
90265 FORMAT('0',I4,' RECORDS WERE EXTRACTED')
      IF(IPRT .EQ. 'YES ')GO TO 50000
50008 IEXT=0
505   WRITE(6,9025,ERR=8000)
510   READ(6,9002,ERR=8000) IOPT
      IF (IOPT.EQ.'YES ') GO TO 200
      IF (IOPT.EQ.'NO  ') GO TO 600
      WRITE(6,9005,ERR=8000)
      GO TO 510
520   WRITE(6,9033,ERR=8004)
      GO TO 505
C     >>>>>>>  TERMINATE PROCESS  <<<<<<<
600   CONTINUE
610   CALL CLOSE(7,ISTAT)
      CALL CLOSE(3,ISTAT)
      RETURN
C     >>>>>>>   ERROR REPORTING   <<<<<<<
C     >>>>>>>  COMMAND DEVICE ERROR  <<<<<<<
8000  WRITE(6,9003,ERR=610)
      GO TO 610
C     >>>>>>>  PRINT FILE ERROR  <<<<<<<
8004  ISTAT=-2
8005  WRITE(6,9004,ERR=8000) ISTAT
      GO TO 610
C     >>>>>>>  DATABASE FILE ERROR OPEN/FORMAT  <<<<<<<
8010  WRITE(6,9012,ERR=8000) ISTAT
      GO TO 610
C     >>>>>>>  DATABASE FILE READ ERROR  <<<<<<<
8015  WRITE(6,9013,ERR=8000) IREC,IPBLK(1)
      GO TO 610
C     >>>>>>>   I/O FORMATTING  <<<<<<<
9000  FORMAT(1H1/1X,'KGS PUBLICATIONS DATABASE'/1X,
     - 'RECORD EXTRACT MODE ',5X,I2,'/',I2,'/',I2,
     - 3X,I2,':',I2,':',I2/1X)
9001  FORMAT(1X,'OPTIONAL PRINTER OUTPUT? (YES OR NO)')
9002  FORMAT(A4)
9003  FORMAT(1X,'COMMAND DEVICE ERROR')
9004  FORMAT(1X,'ERROR ON PRINT FILE STATUS=',I3)
9005  FORMAT(1X,'ILLEGAL COMMAND...RE-ENTER')
9007  FORMAT(1X,'ENTER EXTRACT CRITERIA (AUTHOR OR KEY WORD)')
9008  FORMAT(1X,'ENTER AUTHOR NAME (LAST, FI MI)')
9009  FORMAT(8A4)
9010  FORMAT(6X,'ENTER KEY WORD/S'/1X,
     - 'MAX 6 - 1 PER LINE - UP TO 16 CHAR EACH - "CR" TO STOP'/)
9011  FORMAT(4A4)
9012  FORMAT(1X,'DATABASE FILE ERROR STATUS=',I3)
```

```
9013  FORMAT(1X,'DATABASE READ ERROR AT REC ',I4,2X,
     - 'STATUS=',Z8)
9014  FORMAT(1X,'AUTHOR:              ',8A4,
     - 10X,'RECORD NO. ',I4)
9015  FORMAT(1X,'AUTHOR(2):           ',8A4)
9016  FORMAT(1X,'AUTHOR(3):           ',8A4)
9017  FORMAT(1X,'PUBLICATION DATE:    ',3A4)
9018  FORMAT(1X,'TITLE:               ',13A4)
90181 FORMAT(22X,12A4)
9019  FORMAT(1X,'PUBLICATION NAME:    ',8A4,2X,
     - 'REFERENCE:  ',5A4)
9020  FORMAT(1X,'- - KEY WORDS - -  ',3(4A4,4X))
9021  FORMAT(20X3(4A4,4X))
9022  FORMAT(1X,'AUTHOR EXTRACTED WAS: ',1X,
     - 8A4,1X,'PAGE ',I4)
9023  FORMAT(1X,'KEY WORD/S SELECTED WERE: ',
     - 3(4A4,2X),5X,'PAGE ',I4)
9024  FORMAT(28X,3(4A4,2X))
9025  FORMAT(/1X,'CONTINUE?')
9028  FORMAT(1H1/20X,'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)
9029  FORMAT(1X,'- - - - - - - - - - - - - - - - - - - - -',1X,
     - '- - - - - - - - - - - - - - - - - - - -',1X,
     - '- - - - - - - - - -')
9030  FORMAT(1X,'FILE CODE:           ',4A4)
9031  FORMAT(1X,'ENTER TYPE OF SEARCH..FIELD',1X,
     - 'OR STRING')
9032  FORMAT(1X,'ILLEGAL SEARCH TYPE..RE-ENTER')
9033  FORMAT(1X,'ILLEGAL CHARACTERS IN STRING SEARCH')
9034  FORMAT(A1)
9035  FORMAT(1X,'ENTER "UNION" (OR) OR "INTERSECTION" (AND)',
     - 1X,'FOR SEARCH TYPE')
9036  FORMAT(A3)
9037  FORMAT(1X,'SEARCH CRITERIA USED: ',A4)
      END
KGSEX   FUNC/SUB
.Q      EXT FUNC
.P      EXT FUNC
IPBLK   INT4 VAR
RECORD  INT4 VAR
INAUTH  INT4 VAR
INKEY   INT4 VAR
IDATE   INT4 VAR
ITIME   INT4 VAR
DBNAME  INT4 VAR
IKMAP   INT4 VAR
SKEY    INT4 VAR
ORGST   INT4 VAR
JLU     INT2 VAR
INDEX2  INT2 VAR
IPAGE   INT4 VAR
DATE    EXT FUNC
```

```
TIME    EXT FUNC
OPENW   EXT FUNC
ISTAT   INT4 VAR
8010    LABEL
100     LABEL
9000    LABEL
8000    LABEL
@H      EXT FUNC
9001    LABEL
120     LABEL
9002    LABEL
IPRT    INT4 VAR
140     LABEL
200     LABEL
9005    LABEL
160     LABEL
8005    LABEL
205     LABEL
I       INT4 VAR
215     LABEL
210     LABEL
J       INT4 VAR
212     LABEL
9007    LABEL
220     LABEL
IPAR    INT4 VAR
280     LABEL
250     LABEL
9010    LABEL
ISTYP   INT4 VAR
260     LABEL
9011    LABEL
265     LABEL
300     LABEL
9035    LABEL
270     LABEL
9036    LABEL
9008    LABEL
9009    LABEL
9031    LABEL
8004    LABEL
285     LABEL
ISRC    INT4 VAR
9032    LABEL
IREC    INT4 VAR
IEXT    INT4 VAR
ICRT    INT4 VAR
ILU     INT4 VAR
310     LABEL
9028    LABEL
9029    LABEL
9022    LABEL
```

```
9023    LABEL
J1      INT4 VAR
J2      INT4 VAR
J3      INT4 VAR
9024    LABEL
J4      INT4 VAR
J5      INT4 VAR
J6      INT4 VAR
9037    LABEL
IRNUM   INT4 VAR
340     LABEL
SYSIO   EXT FUNC
500     LABEL
IOR     INT4 VAR
370     LABEL
400     LABEL
350     LABEL
342     LABEL
STSRCH  EXT FUNC
520     LABEL
II      INT4 VAR
344     LABEL
346     LABEL
360     LABEL
9014    LABEL
9015    LABEL
9016    LABEL
9030    LABEL
K1      INT4 VAR
9017    LABEL
9018    LABEL
90181   LABEL
9019    LABEL
9020    LABEL
K2      INT4 VAR
K3      INT4 VAR
9021    LABEL
K4      INT4 VAR
K5      INT4 VAR
K6      INT4 VAR
372     LABEL
9034    LABEL
IIDUM   INT4 VAR
405     LABEL
410     LABEL
420     LABEL
IKEY    INT4 VAR
430     LABEL
435     LABEL
IKST    INT4 VAR
IKSP    INT4 VAR
438     LABEL
```

```
440      LABEL
445      LABEL
450      LABEL
462      LABEL
JJ       INT4 VAR
463      LABEL
KK       INT4 VAR
465      LABEL
470      LABEL
480      LABEL
475      LABEL
478      LABEL
IKCNT    INT4 VAR
IHIT     INT4 VAR
495      LABEL
498      LABEL
8015     LABEL
50000    LABEL
50008    LABEL
50001    LABEL
50002    LABEL
90261    LABEL
50003    LABEL
90262    LABEL
50004    LABEL
50005    LABEL
90263    LABEL
50007    LABEL
90264    LABEL
90265    LABEL
505      LABEL
9025     LABEL
510      LABEL
IOPT     INT4 VAR
600      LABEL
9033     LABEL
610      LABEL
CLOSE    EXT FUNC
9003     LABEL
9004     LABEL
9012     LABEL
9013     LABEL

   0 ERRORS: FORTRAN V LEVEL 1 R04-00
```

```
      SUBROUTINE STSRCH(ORGST,INST,ISTAT)
C     PROGRAM DESIGNED TO SEARCH AN ASCII
C     FIELD AGAINST AN INPUT FIELD TO
C     DETERMINE A MATCH CONDITION.
C     WRITTEN BY L. KAETZEL - CBT - 02/23/79
C     >>>>>>>  ARGUMENT DEFINITIONS  <<<<<<<<<
C     ORGST = ORGINAL STRING MAX 100 CHARACTERS
C     INST = INPUT STRING MAX 100 CHARACTERS
C     STAT = DISPOSITION OF PROCESSING
C          0=NO MATCH
C          1=MATCH
C         -2=NON ASCII CHARACTERS IN STRING
      INTEGER ORGST(25),INST(25)
      INTEGER OSTOR,ISTOR,OSCT,TCCT,BLNK
C     >>>>>>>  FIND NO. OF CHAR FOR SEARCH  <<<<<<<
      NCHAR=-1
      ITMP='1111'
      DO 30 I=1,25
      DO 20 J=1,4
      ISCON=32-(8*J)
      ICHAR=ISHFT(INST(I),-ISCON)
      ISTOR='    '
      CALL ISBYTE(ICHAR,ISTOR,0)
      IF (ISTOR.EQ.'    '.AND.ITMP.EQ.'    ')
     - GO TO 40
      NCHAR=NCHAR+1
      ITMP=ISTOR
20    CONTINUE
30    CONTINUE
C     >>>>>>>  VALIDATE ASCII STRINGS  <<<<<<<
40    ICOUNT=0
      NDIF=(NCHAR/4)+1
      DO 100 I=1,25
      DO 50 J=1,4
      ISCON=32-(8*J)
      ICHAR=ISHFT(ORGST(I),-ISCON)
      ISTOR='    '
      CALL ISBYTE(ICHAR,ISTOR,0)
      IF (ISTOR.LT.Y'20202020'.OR.
     - ISTOR.GT.Y'5A202020') GO TO 600
      ICHAR=ISHFT(INST(I),-ISCON)
      ISTOR='    '
      CALL ISBYTE(ICHAR,ISTOR,0)
      IF (ISTOR.LT.Y'20202020'.OR.
     - ISTOR.GT.Y'5A202020') GO TO 600
      ICOUNT=ICOUNT+1
      IF (ICOUNT.EQ.NCHAR) GO TO 150
50    CONTINUE
100   CONTINUE
C     >>>>>>>  MASK IT  <<<<<<<
150   OSCT=1
      ISCT=1
```

```
      TCCT=1
      II=1
      I=1
      ISTMP='AAAA'
200   ISCON=32-(8*I)
      IOCHAR=ISHFT(ORGST(OSCT),-ISCON)
      IOSTOR='    '
      CALL ISBYTE(IOCHAR,IOSTOR,0)
      ISCON=32-(8*II)
      IF (IOSTOR.EQ.'    '.AND.ISTMP.EQ.'    ') GO TO 275
      ISTMP=IOSTOR
      IICHAR=ISHFT(INST(ISCT),-ISCON)
      IISTOR='    '
      CALL ISBYTE(IICHAR,IISTOR,0)
      IF (IOSTOR.NE.IISTOR) GO TO 300
C     >>>>>>>  CHARACTER MATCH  <<<<<<<
      TCCT=TCCT+1
      IF (TCCT.GT.NCHAR) GO TO 500
      II=II+1
215   I=I+1
      IF (II.LE.4.AND.I.LE.4) GO TO 200
      IF (II.LE.4) GO TO 250
      II=1
      ISCT=ISCT+1
      IF (ISCT.LE.25) GO TO 250
      ISTAT=1
      RETURN
250   IF (I.LE.4) GO TO 200
      I=1
      OSCT=OSCT+1
      IF (OSCT.LE.25) GO TO 200
275   ISTAT=0
      RETURN
C     >>>>>>>  CHARACTER NO-MATCH  <<<<<<<
300   ISCT=1
      II=1
      TCCT=1
      GO TO 215
500   ISTAT=1
      RETURN
600   ISTAT=-2
      RETURN
      END
STSRCH  FUNC/SUB
.Q      EXT FUNC
.P      EXT FUNC
ORGST   FORM PAR
INST    FORM PAR
ISTAT   FORM PAR
OSTOR   INT4 VAR
ISTOR   INT4 VAR
OSCT    INT4 VAR
```

```
TCCT     INT4 VAR
BLNK     INT4 VAR
NCHAR    INT4 VAR
ITMP     INT4 VAR
30       LABEL
I        INT4 VAR
20       LABEL
J        INT4 VAR
ISCON    INT4 VAR
ICHAR    INT4 VAR
ISHFT    EXT FUNC
ISBYTE   EXT FUNC
40       LABEL
ICOUNT   INT4 VAR
NDIF     INT4 VAR
100      LABEL
50       LABEL
600      LABEL
150      LABEL
ISCT     INT4 VAR
II       INT4 VAR
ISTMP    INT4 VAR
200      LABEL
IOCHAR   INT4 VAR
IOSTOR   INT4 VAR
275      LABEL
IICHAR   INT4 VAR
IISTOR   INT4 VAR
300      LABEL
500      LABEL
215      LABEL
250      LABEL

   0 ERRORS: FORTRAN V LEVEL 1 R04-00
```

APPENDIX E: KGS EXTERNAL SUBROUTINES

Subroutine	Function
SYSIO	Performs input/output
CFILW	Create mass storage files
DFILW	Deletes mass storage files
OPENW	Opens mass storage files or devices
CLOSE	Closes mass storage files or devices
IFETCH	Gets program overlay from disk
DATE	Gets system date
TIME	Gets system time
ISHFT	Shifts memory location left or right a specified number of bits
ILBYTE	Loads a specified byte
ISBYTE	Stores a specified byte

APPENDIX F: KGS ERROR DIAGNOSTICS

Data Base File Error, Status=YYYY
Error Opening Overlay For Mode XXXX, Status=YYYY
Error Fetching Overlay For Mode XXXX, Status=YYYY

Where XXXX= KGS run mode
YYYY= Return status from program library subroutine

Status Value	Meaning
1	Illegal function, illegal file type
2	LU error, illegal logical unit
3	Volume error, no such volume
4	Name error, no such file name
5	Size error, no room on disk
6	Protect error, mismatch on protect keys
7	Privilege error
8	Buffer error, no room for file control block
9	Assignment error, logical unit already assigned
11	File descriptor syntax error
112-225	File input or output error

FEDERAL INFORMATION PROCESSING STANDARD SOFTWARE SUMMARY

01. Summary date: Yr. 79, Mo. 11, Day 26

02. Summary prepared by (Name and Phone): Lawrence Kaetzel 921-3229

03. Summary action: New [X] Replacement [ ] Deletion [ ]
Previous Internal Software ID:

04. Software date: Yr. 79, Mo. 11, Day 26

05. Software title: KGS Data Base System

06. Short title: KGS

07. Internal Software ID: PDB

08. Software type:
- [ ] Automated Data System
- [X] Computer Program
- [ ] Subroutine/Module

09. Processing mode:
- [X] Interactive
- [ ] Batch
- [ ] Combination

10. Application area

General:
- [ ] Computer Systems Support/Utility
- [ ] Scientific/Engineering
- [X] Bibliographic/Textual
- [ ] Management/Business
- [ ] Process Control
- [ ] Other

Specific: Scientific lab or office

11. Submitting organization and address:
National Bureau of Standards
National Engineering Laboratory
Center for Building Technology
Bldg. 226, Room B 266
Washington, D.C. 20234

12. Technical contact(s) and phone:
Lawrence Kaetzel
Robert Glass

13. Narrative:
The KGS data base system allows the indexing and retrieval of scientific papers through the use of a minicomputer system in an interactive mode. Criteria are entered through the user's terminal which produces subsets of the data base in a report format as well as statistical summaries of data base elements.

14. Keywords:
data base, information retrieval, interactive processing

15. Computer manuf'r and model	16. Computer operating system	17. Programing language(s)	18. Number of source program statements
Perkin-Elmer 7/32	OS32MT4.3	FORTRAN	1,359

19. Computer memory requirements	20. Tape drives	21. Disk/Drum units	22. Terminals
2,375 thousand bytes	3	7	20

23. Other operational requirements:

24. Software availability
- Available [X]
- Limited [ ]
- In-house only [ ]

25. Documentation availability
- Available [X]
- Inadequate [ ]
- In-house only [ ]

26. FOR SUBMITTING ORGANIZATION USE

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

U.S. DEPT. OF COMM. BIBLIOGRAPHIC DATA SHEET (See instructions)	1. PUBLICATION OR REPORT NO. NBS TN 1123	2. Performing Organ. Report No.	3. Publication Date October 1980

4. TITLE AND SUBTITLE

A Computer Data Base System for Indexing Research Papers

5. AUTHOR(S) Lawrence J. Kaetzel, Robert A. Glass and 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 above.

10. SUPPLEMENTARY NOTES

[X] 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)

The KGS data base system allows the indexing and retrieval of scientific research papers through the use of a minicomputer system in an interactive mode. Criteria are entered through the user's computer terminal which produces subsets of the data base in a report format as well as statistical summaries of data base elements.

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

Data base; information retrieval; interactive processing.

13. AVAILABILITY

[X] Unlimited

[ ] For Official Distribution. Do Not Release to NTIS

[X] 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
90

15. Price
$4.25

☆ U. S. GOVERNMENT PRINTING OFFICE : 1980 311-046/254

USCOMM-DC 6043-P80

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 $13; foreign $16.25. Single copy, $3 domestic; $3.75 foreign.

NOTE: The Journal was formerly published in two sections: Section A "Physics and Chemistry" and Section B "Mathematical Sciences."

DIMENSIONS/NBS—This monthly magazine is published to inform scientists, engineers, business and industry leaders, teachers, students, and consumers of the latest advances in science and technology, with primary emphasis on work at NBS. The magazine highlights and reviews such issues as energy research, fire protection, building technology, metric conversion, pollution abatement, health and safety, and consumer product performance. In addition, it reports the results of Bureau programs in measurement standards and techniques, properties of matter and materials, engineering standards and services, instrumentation, and automatic data processing. Annual subscription: domestic $11; foreign $13.75.

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.

BIBLIOGRAPHIC SUBSCRIPTION SERVICES

The following current-awareness and literature-survey bibliographies are issued periodically by the Bureau:

Cryogenic Data Center Current Awareness Service. A literature survey issued biweekly. Annual subscription: domestic $35; foreign $45.

Liquefied Natural Gas. A literature survey issued quarterly. Annual subscription: $30.

Superconducting Devices and Materials. A literature survey issued quarterly. Annual subscription: $45. Please send subscription orders and remittances for the preceding bibliographic services to the National Bureau of Standards, Cryogenic Data Center (736) Boulder, CO 80303.

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

U.S.MAIL

SPECIAL FOURTH-CLASS RATE
BOOK