AN INTERACTIVE LITERATURE COLLECTION AND REFERENCE RETRIEVAL SYSTEM

by

YU-JEN EUGENE TSENG

B.A., NATIONAL CHENGCHI UNIVERSITY, 1966

A MASTER'S REPORT

submitted in partial fulfillment of the

requirements for the degree

MASTER OF SCIENCE

Department of Computer Science

KANSAS STATE UNIVERSITY
Manhattan, Kansas

1974

Approved by:

William J. Hankley
Major Professor

PREFACE

This report is meant to serve needs of different readers. It can be read somewhat independently. Chapter I is the introduction. Technical overview is in Chapter II, and Section III.1. A user's guide is in Chapter III, Appendix II.A, and Appendix II.B. Appendix III provides APL instructions of ILCRS for users who have no knowledge of APL. Technical details are in Section III.2, Section III.4, Appendix I, and Appendix II. Demonstration Examples are in Appendix IV.

TABLE OF CONTENTS

	page
PREFACE	ii
LIST OF FIGURES	v
I. INTRODUCTION	1
II. GENERAL SYSTEM DESCRIPTION OF ILCRS	3
II.1 File Organization and Data Structures	3
II.2 Indexing	7
II.3 Capabilities of The System	10
III. USER ORIENTATION AND GUIDE	12
III.1 Introduction	12
III.2 Structures and Operations of User's Workspaces and Function Programs Operations in Workspace	15
III.3 Type of User	20
III.4 Procedures of Using the System	20
1. System initialization	21
2. Data entry	23
3. Data display and editing	27
4. Data storage	29
5. Document retrieval	29
6. Document deletion	35
7. Physical file maintenance	35
8. File initialization	37
III.5 Procedure Summary	38

APPENDICES

	page
APPENDIX I. DETAILS OF DATA STRUCTURES USING APL	43
APPENDIX II. ORGANIZATION OF COMPUTER PROGRAMS	56
A. Definition of Each Function Program	56
B. Definition of Each Group Name	58
C. Algorithms	59
D. List of Groups and Functions	89
APPENDIX III. APL INSTRUCTIONS FOR USING ILCRS	111
APPENDIX IV. DEMONSTRATION EXAMPLES	114
A. Physical File Maintenance Examples	114
B. Operation Examples	114
ACKNOWLEDGEMENTS	137

LIST OF FIGURES

		page
Figure II-1	Data Structures of ILCRS	6
Figure III-0	Data Flow and Library Structures of ILCRS	14
Figure III-1	Structures and Organization of User's Workspaces	16
Figure III-2	Structures and Operations of Groups and Functions in Workspace Ws	17
Figure III-3	Structures and Operations of Groups and Functions in Workspace Ws+1	18
Figure III-4	Structures and Functions and Groups in FNS	19
Figure III-5	Algorithm for System Initialization	22
Figure III-6	Algorithm for Storing Data	24
Figure III-7	Algorithm for Search Request	31
Figure III-8	Format for List of a Document	32
Figure III-9	Algorithm for Deleting a Document	36
Figure III-10	Algorithm for File Initialization	37
Figure A-I-1	Text Structures	44
Figure A-I-2	Structures of Names	50
Figure A-I-3	Structures of Keywords	51
Figure A-I-4	Structures of Surrogate Files	52
Figure A-I-5	General Description of File Organization and Data Structures Using APL	53
Figure A-II-1 to Figure A-II-28	Algorithms	61 to 88

Chapter I

INTRODUCTION

The Interactive Literature Collection and Reference Retrieval System(abbreviated as ILCRS) is an information retrieval system to achieve a user-machine interface in computer reference searching of documents. The reason for building this system is as a pedagogical exercise to program an interactive information retrieval system, with the objective of demonstrating the organization of a total system. The main purpose of maintaining this system is to provide reference retrieval by author's name, name of document, keyword, key phrase, or document number.

ILCRS is designed to meet the demand for fast retrieval of information and for fast maintenance transactions. It is useful in cases where fast response is needed for small volumes of retrieved data but it is generally more expensive than a batch system. A batch system is one that collects user queries and maintenance transactions and runs them whenever convenience and economy permit. For system users, this has a tendency to mean a long wait for results. ILCRS has no middle men and no waits for a batch to be processed. A query is submitted directly to the computer through an on-line terminal, search operations and maintenance transactions can begin almost immediately.

Because the time factor is of crucial importance in searching, this system is developed as an on-line information retrieval system, its goal being the provision of direct and immediate access to the stored data. Users may sit down at any access terminal, make a request, and have their results printed back to them at the terminal in a short period of time. The user also can add a new document into the file or delete a document from the file at any convenient time. Although actual file searching is no faster in an interactive system, the requestor may be looking at the first retrieved record while the computer searches for the next one. Hence from his point of view, the operation seems faster.

The system data bases include not only journals, reports, theses, articles, conferences, proceedings, books but also personnel data, and any other documents which will become easier to access, if stored in the system. The storage allocation for the system is flexible and extensible.

Chapter II

GENERAL SYSTEM DESCRIPTION

The major guiding principle in the conception and design of the ILCRS system is to produce a user-oriented system. It is assumed that in a user-oriented system the user should not be expected to know about the technicalities of the system, that he should not be forced to express his request in very restricted and structured vocabulary form, and that his interface with the system should be through very simple commands.

This chapter gives a brief overview of the system which was built. Further details are in Appendices and Chapter III.

II.1 File Organization and Data Structures

A file is a set of records with some structural elements and some semantic attributes in common. Just as a character is built up from bits, a field from characters, and a record from fields, a file is created by grouping records according to some organizational rules.

A file is said to be organized in that it consists of records which almost always are placed in sequence, according to some defined criterion, and that the records consist of fields organized to form the records. For any given system a file organization is completely specified if we know the

following:

1. The data, or set of fields, presented in the records of the file
2. The sequence of records within the file and physical placement of reocrds within the file
3. The characteristics of storage media in which the file is stored

There are two principal file organizations used in ILCRS, known as the direct and inverted systems. In the direct system, the file is stored in order by document numbers, and intems to be retrieved are identified by a sequential scan of the complete file. The second major system of file organization known as the inverted system is based on an arrangement of information file in order by the main information identifiers, normally a set of key words or index terms. Both the direct and the inverted file systems suffer from the fact that in many cases a rather substantial part of the file must be examined before it becomes possible to isolate the items to be compared with the search request, whereas in the inverted system, the number of required comparsions depends on the number of terms specified in the search request, more particularly, the more terms specified in the search request, the larger the size of the file that must be examined, even though, in general, the expected number of documents to be retrieved decreases. In this situation, ILCRS system trades search time for storage space by making use of

a dual file system consisting first of an inverted index file which provides for each acceptable index term a list of references to the documents indexed by that particular term, plus a direct file containing the reference information for all documents and some surrogate file index to authors, keywords, and owners. The ILCRS system, like the standard inverted system, requires the construction and maintenance of a large auxilary index file. Conceptually, there are five basic information structures, with the cross reference as shown in Figure II-1. More details are in Appendix I.

1. Author file -- with each author's name indexed to document number(s). The storage of names is automatically done by the key transformation of the system.
2. Title and text information file -- are in sequential order with an index to author(s), an index to document owner(s), and an index to keywords.
3. Keyword file -- with each keyword(key phrase) indexed to document number(s). The storage of keywords is automatically done by the key transformation of the system.
4. Owner file -- sequential list of owner's names.
5. Course reading list file -- in sequential order with index to document numbers.

Author File (in alphabetical order)

list of names	index to document numbers
AHO, A.	9 10
BENSON, G.	19

Text Information File
(sequential)

Text Information	index to keywords	index to authors	index to owners
COBOL PROGRAMMING,'71 JOHN WILEY & SONS,INC	21	96 17	2 5

Keyword File (in alphabetical order)

list of key words	index to document numbers
AUTOMATA	11

Course Reading List File

list of course names	index to document numbers
CS609	4 15 9 20
CS400	10 37 8 56

Owners File

list of owners
FISHER
STERN

Figure-II-I

All the files of ILCRS are stored on disk for fast retrieval and fast maintenance.

II.2 Indexing

With the entered document data, the indexing procedures are automatically done by the system. The document data are:

1. Authors (or names) are recorded, last name first, followed by initial of first name. In case of a collection, symposium, conference, etc., the author(s) name(s) are used. When no author is indicated the issuing organization or identified name is used. usually an issuing organization is indicated by standard abbreviations.
2. Title and text

   Title is recorded as it appears on the title page or with a standard abbreviation. Any other text information shall be followed line by line.

   a. Bibliographic reference:

      a1). Document number -- automatically assigned

      a2). Citation:

      Book: place and publisher, if a portion of a a book, add inclusive pages. For symposiums, conferences, etc., record name of meeting, sponsoring organization, place, date held, publisher.

      a3). Report: place and name of issuing agency,

including report number, project name and number, and month of issue.

a4). Journal article: title of periodical, volume number, inclusive pages, and date(see American Standard for Periodical Title abbreviations ASA Z39.5-1963 for abbreviations).

b. Journal: American standard for Periodical Title abbreviations are used, publisher, and volume number.

c. Abstract: abstract of a document.

d. Any other information capable of being stored line by line.

3. Key words(Key phrases)

The indexer will decide which terms or phrases best describe or give clues to the document content. The descriptors include subject headings, key words, and short key phrases. The descriptors are created by the users using primarily the vocabulary and phrasing of the author. The advantage of using free descriptors, is how easy it makes for users to get started, which increases their confidence in using the system. Usually a user phrasing a subject request will provide some results on the subject because alternative words and phrasing used by the author are picked up by the indexers. Therefore the highly

relevant documents on a given subject can be retrieved by any of several differently worded user requests. Some of the users come to the system not to get all of the information about a subject but only to get a few good documents about it. For this purpose—as well as for complete coverage—free vocabulary, in-depth, indexing is excellent.

4. Owner

Owner actually represents location, sponsor, indexer, or owner of a document.

II.3 Capabilities of The System

The system is designed for fast information storage and retrieval. Document information can be entered and deleted at the user's convenience. There is no waiting for a batch to be processed. The users may sit down at any of the terminals, make a request, and have their data stored, deleted, or retrieved. The capabilities of the system can be classified into the following:

1). Storage of document data -- After the system has been generated, data should be stored in appropriate location for future search requests.

2). Deletion of document data -- to update the data bases in the system.

3). Garbage collection -- collects the available storage locations (not invoked automatically: only invoked by user request).

4). Document retrieval by name -- to retrieve a specific document, or documents written by a given author.

5). Document retrieval by keyword(key phrase, or subject)-- to identify a document or documents corresponding to a specific keyword(keyphrase).

6). Document retrieval by document number(s) -- to retrieve specific information or to locate document(s) already known to exist.

7). Document retrieval by keyword number(s) -- to identify

a document or documents corresponding to one or more keywords known to exist(indirect -- via keyword cross references).

8). Course reading list retrieval -- to search references and readings for a course.

Chapter III

USER ORIENTATION AND GUIDE

III.1 Introduction

The ILCRS system is a fully automatic document storage, deletion, and retrieval system, capable of processing search request and storing data. The system program functions were written in APL.

This chapter provides information necessary to use the ILCRS system. It describes procedures required for interacting with the system: how to access the data bases, how to retrieve data, and how to create and delete data sets.

All the data bases of ILCRS are stored in the user's private library. The library contains N files, where each file has two workspaces named Ws and Ws+1 (s=1,3,..,2n-1., N is the number of files in the user's private library). When a document is to be stored, the user must decide where the data is to go, that is, on which file the document information will be stored. For example, if the user classified a document into x th file, he should load two workspaces of the x th file—W2x-1 and W2x into the active workspace, separately. After the formalized data has been entered into the active workspace, it must be saved. To delete a document from the file, the user also has to load the workspaces of

the file, delete information from the workspaces and save the workspaces. To retrieve the information, the user can either load the workspace where the data were stored, or copy the data from his or other user's library into the active workspace, then start search requests.

For different purpose of operations, the user should have different function program(or group) copied into his active workspace and executed in the active workspace.

APL instructions for ILCRS user are in APPENDIX III.

Figure III-0 represents the flow of data file, function program, and group between user and the computer. It summarizes the following points:

a. The user can see or use only programs, groups, or data that are in user's currently active workspace.
b. The user can save what is in his active workspace into his private library.
c. The user can load into his active area from his own library, or from another user's .
d. The user can copy function program, group, or data into his active area from his own library or from another user's.
e. The user can erase function program, group, or data from his active workspace to avoid cluttering up the workspace.

Data Flow and Library Structures of ILCRS

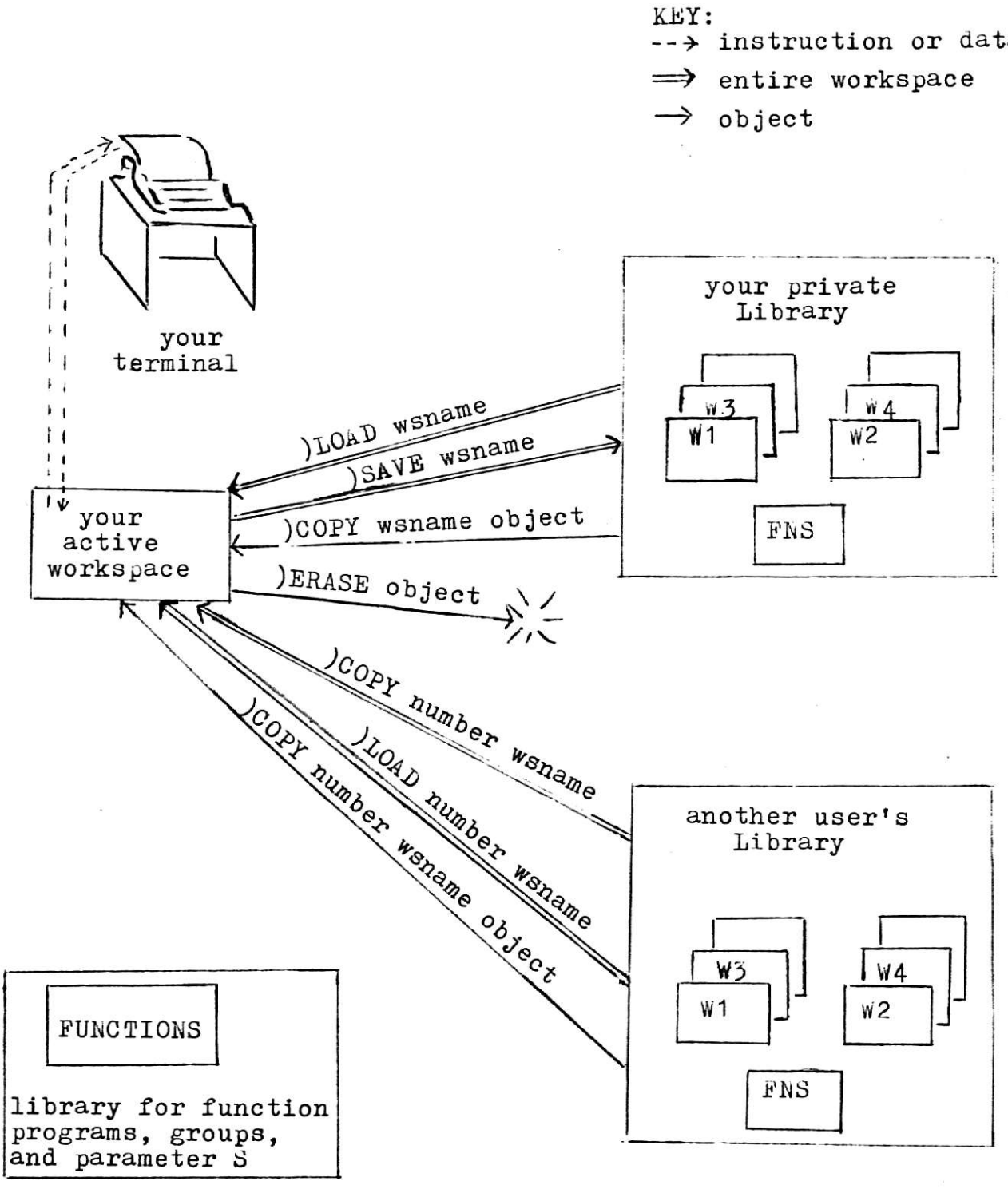


Figure-III-0

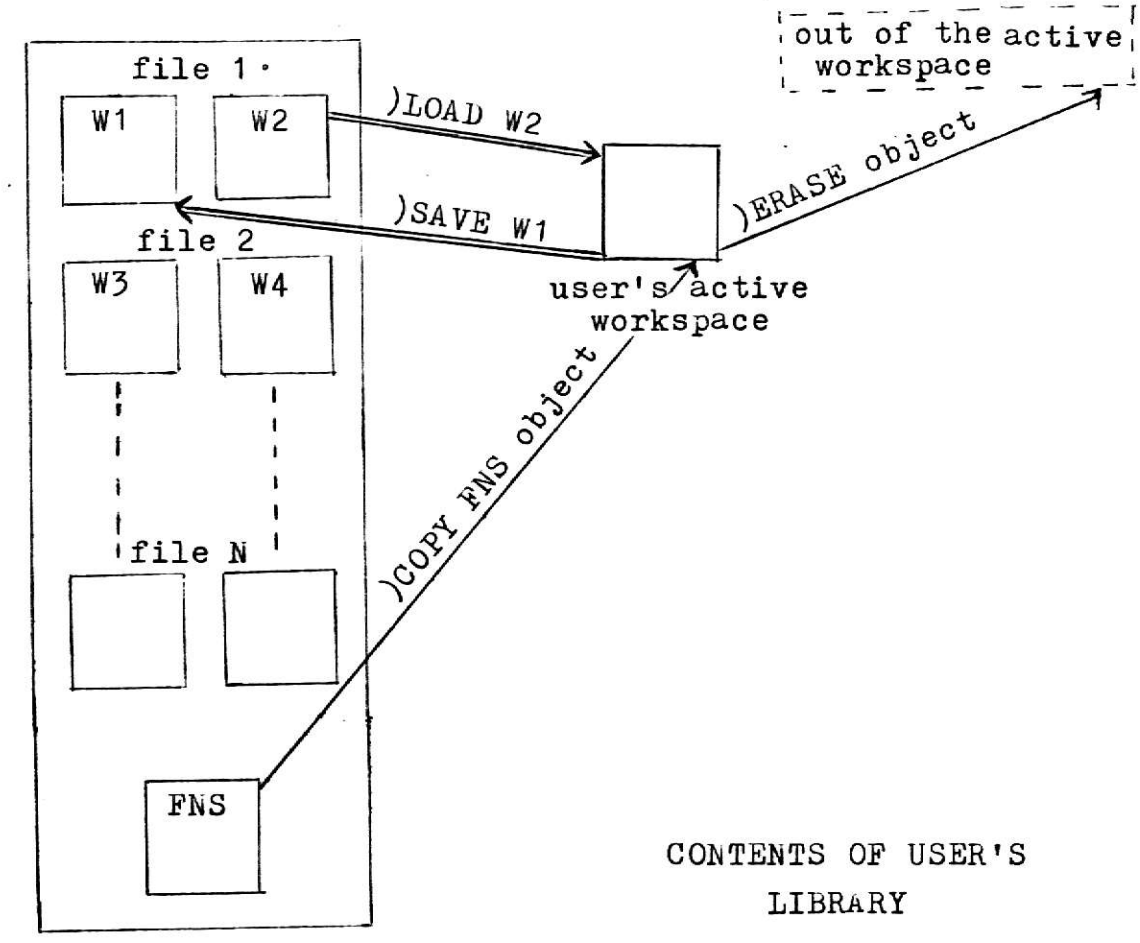
III.2 Structures and Operations of User's Workspaces and Function Program Operations in Workspaces

The user can create as many files as he needs (see Figure III-1). Each file has two workspaces—Ws and Ws+1 (s=1,3,.., 2n-1). Ws is used to store records of authors' names, text information derived from the documents and owners of each document (see Figure III-2). Ws+1 is used to store records of descriptors (see Figure III-3).

When the user is requesting storage or search, he should access Ws except for retrieval by keyword (key phrase) or keyword (key phrase) number(s) (where Ws+1 has to be accessed). All the program functions are stored in the workspace FNS (see Figure III-4), so if the requested function is not in the current active workspace, the user should enter:)COPY FNS object .

Any of program functions can be erased or be copied back in the workspace Ws or Ws+1.

Sturctures and Organization of
User's Workspaces



KEY:

⟹ entire Workspace

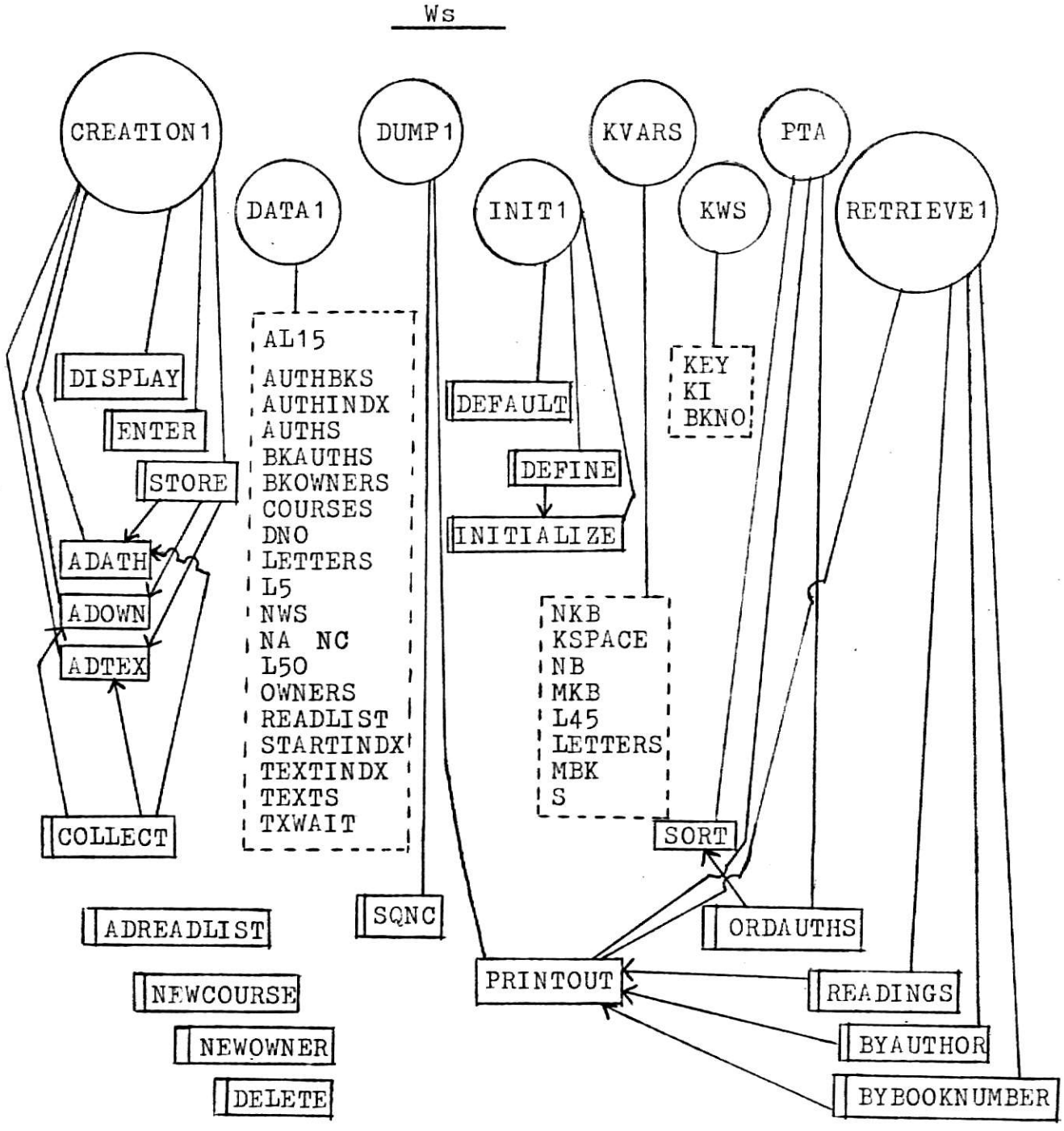
⟶ object

Figure III-1

CONTENTS OF USER'S LIBRARY

workspace name	contents
Ws s=1,3,--,2n-1	1.name, text, and owner data bases 2.grouped variables 3. may have some functions saved
Ws+1 s=1,3,--,2n-1	1. keyword data bases 2. grouped variables 3. may have some fundtions saved
FNS	all the functions, groups, and parameter--S

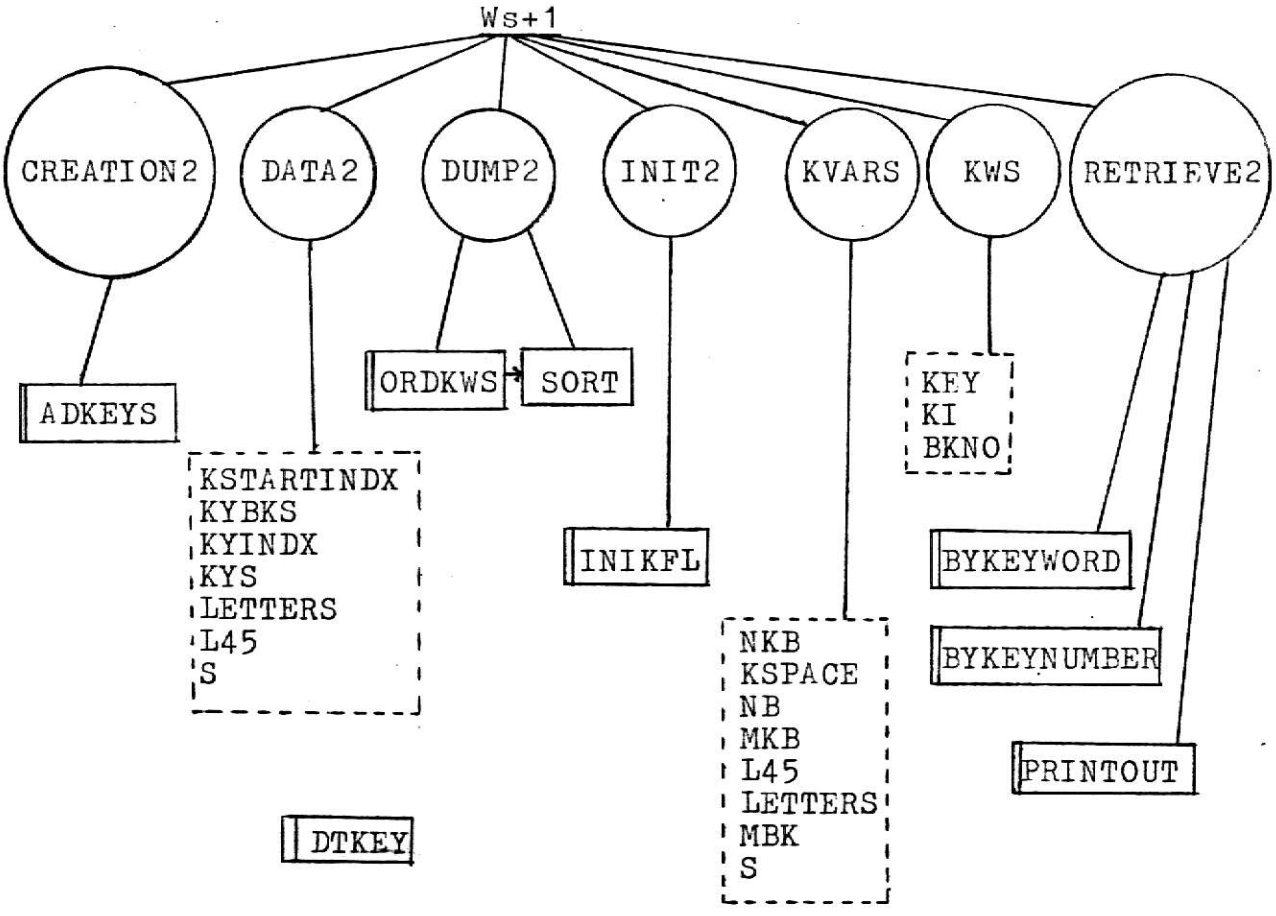
Structures and Operations of Groups and Functions
in Workspace Ws



KEY: see next page

Figure III-2

Structures and Operations of Groups and Functions
in Workspace Ws+1



KEY:

Group name

Function name not executable by user

Call a Function

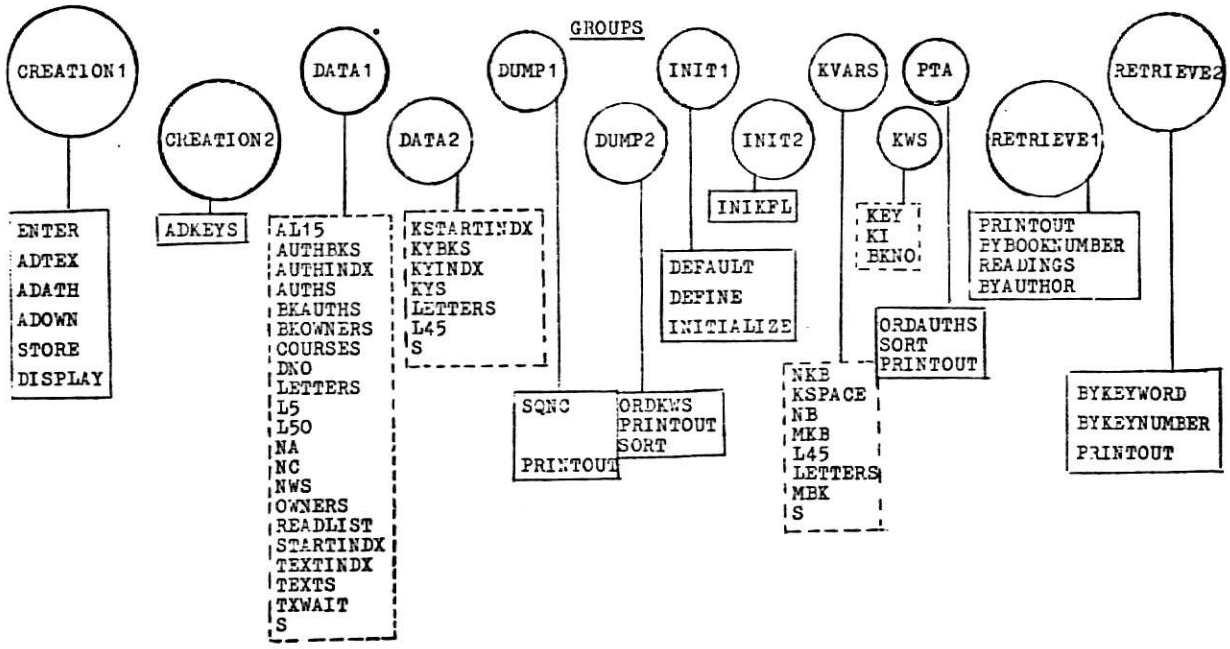
Function name executable by user

Variables

Explanation of Functions
are in Appendix II

Figure III-3

Structures of Functions and Groups in FNS



OTHER FUNCTIONS

ADREADLIST COLLECT DELETE NEWCOURSE NEWOWNER SYSGEN

DTKEY

VARIABLES

S

KEY:

Group name

Function name

Variable

Explanation of names is in Appendices I & II.

Figure-III-4

III.3 Type of User

The system users can be divided into three levels of users:

(1). System manager who is responsible for the system initialization, file creation, and system maintenance.

(2). The regular users who are interested in information storage and search request procedures.

(3). Any other users who are only interested in doing the search request.

III.4 Procedures of Using the System

The procedures of using the system can be classified into :

1. System initialization
2. Data entry
3. Data display and editing
4. Data Storage
5. Document retrieval
6. Document deletion
7. Physical file maintenance
8. File initialization

1. System initialization (see Figure III-5 for algorithm)

The system must be initialized before it can store the data base and search the query for the user. The initialization steps are:

a. Load all the system's program functions from the library and create workspaces:

All the program functions of the system were saved in the library. When system is to be initialized the user should copy that into the workspace named FNS and save. The steps are:

1). Enter:)LOAD library-number FUNCTIONS

2). Enter:)WSID FNS

3). Enter: SYSGEN

4). Follow the instructions returned by the system

b. Define the values for each parameter by following the instructions of the system. The system will tell which parameter's value is to be typed in. In order to prevent wasting of time, the user should have the value of each parameter in hand, so he can respond to the system immediately. In other words, the system initializer is supposed to work out the value of each parameter before he can start to use the system. It is necessary for the initializer to read the statements of function named 'DEFINE' and 'DEFAULT' (see Appendix II Section D).

Algorithm for System Initialization

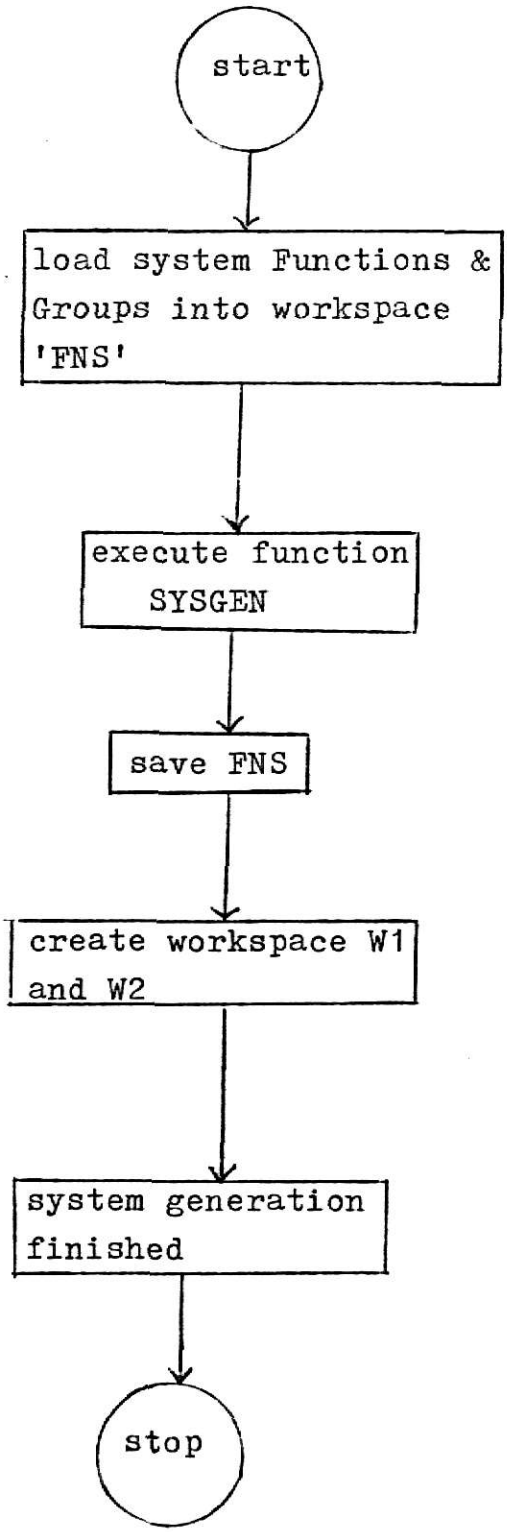


Figure III-5

c. If the defining the value of each parameter is not necessary, the user should enter: DEFAULT . After the execution of the function, DEFAULT, the user can redefine some of the parameters in the system. The default value of each parameter is described in the program function 'DEFAULT' (see Section D, Appendix II).

2. Data entry (see Figure III-6 for algorithm)

A. Enter a document

The necessary steps are:

a. Load Ws into active workspace by typing:
)LOAD Ws (s=1,3,...,2n-1)

b. Copy the function group 'CREATION1' into Ws by typing (as needed):)COPY FNS CREATION1

c. Execute the function 'ENTER' by typing: ENTER

d. Start entering the data

1). Format for the string consisting of the author's name (publisher, identifier of document, or name of organization recorded by standard abbreviation):

'A_last name,ßX.' or 'A_letters', where ß is blank, X is initial of first name. If there is no author, enter : A_ .

Example 1:

If a book has two authors named 'Gray Brown' and 'Willam Crouse' then the user shall type in:

ALGORITHM FOR STORING DATA

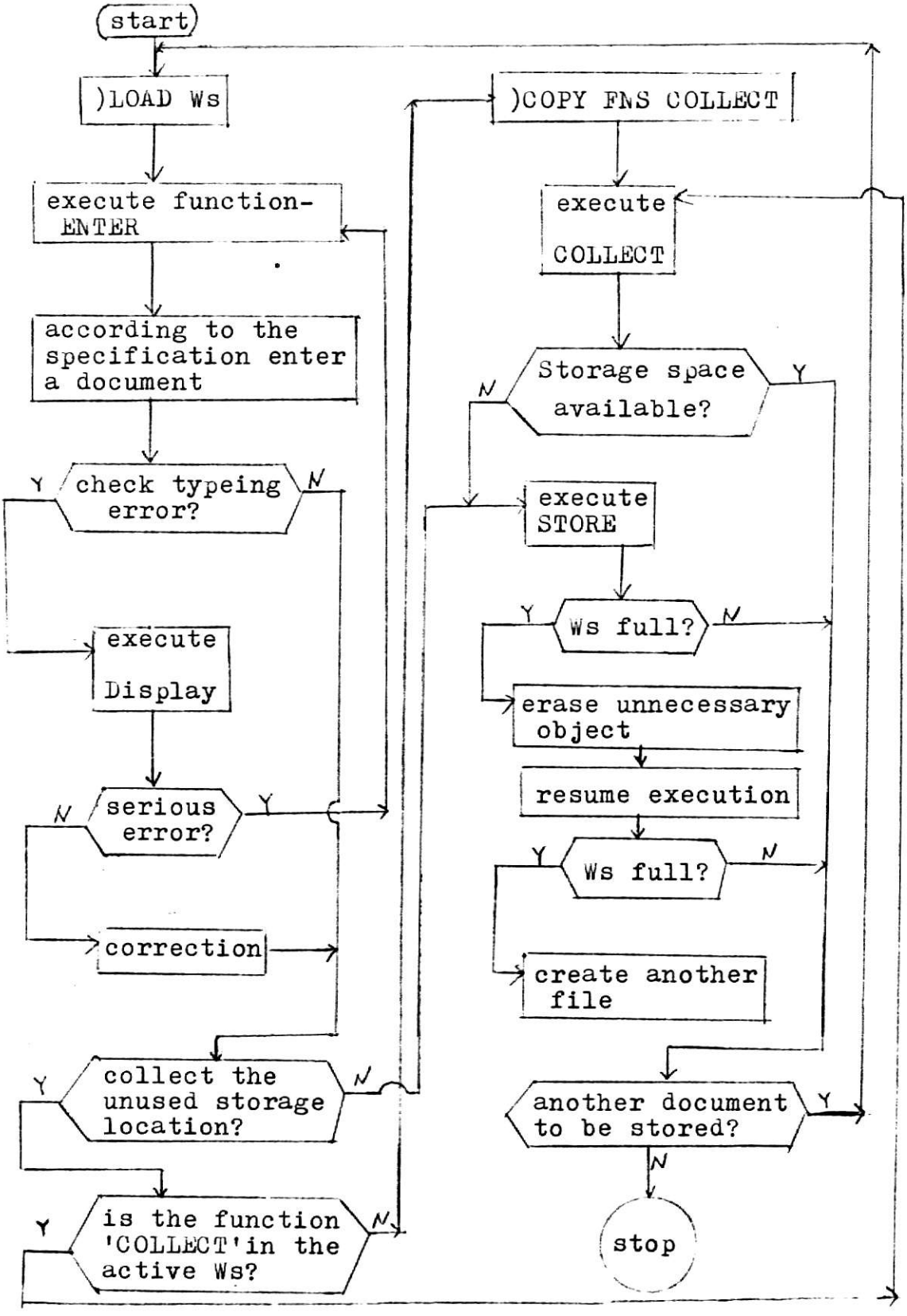


Figure-III-6

A_BROWN, G.

A_CROUSE, W.

Example 2:

A_ACM

or

A_IEE

2). Text information format

T_followed by at most L50 (L50 is a parameter defined in system initialization procedure, the default value is 65.) characters for each line.

Examples:

T_SYSTEM ANALYSIS FOR DATA TRANSMISSION, PRENTICE

T_HALL, ENGLEWOOD CLIFFS, N. J., 1972, P.909

or

T_VOL. 1, VOL. 2, VOL. 3, VOL. 4

or

T_//JOBNAME JOB (STANDARD JOB INFORMATION)

T_//STEPNAME EXEC STATPROG,ROUTINE=NONPAR

T_//STATPROG.SYSIN DD *

T_-----PARAMETER AND DATA CARDS------

T_/*

3). Descriptor format

The maximum number of descriptors for a document is MKB (MKB is a parameter defined in system initialization procedure, the default value is 8.).

K_followed by descriptor (keyword)

If there is no keyword, enter:

K_

Example:

K_PROGRAMMING LANGUAGES

K_PARSING

4). Owner format

The maximum number of owners for a document is OL3 (OL3 is a parameter defined in system initialization procedure, the default value is 3.).

Example:

O_FISHER

O_CALHOUN

5). End of entering

After all the required information is entered, the user shall type in : END .

B. Add a new owner's name into the file

The length of a owner's name was defined in the system initialization procedure. The default length is up to 15.

Enter the following commands as needed:

1).)LOAD Ws (Access to workspace Ws, s=1,3,..,2n-1)

2).)COPY FNS NEWOWNER

3). NEWOWNER

4).)ERASE NEWOWNER

5).)SAVE Ws

C. Add a new course name to the file

The length of a course name was defined in system initialization procedure. The default length of a course name is up to 5.

Enter the following commands as needed:

1).)LOAD Ws (s=1,3,..,2n-1)

2).)COPY FNS NEWCOURSE

3). NEWCOURSE

4).)ERASE NEWCOURSE

5).)SAVE Ws

D. Store the reading list for each course

Enter the following commands as needed:

1).)LOAD Ws (s=1,3,..,2n-1)

2).)COPY FNS ADREADLIST

3). ADREADLIST (A list of course names can be seen, if 'COURSES' was typed before the execution of 'ADREADLIST'.)

4).)ERASE ADREADLIST

5).)SAVE Ws

3. Data display and editing

To check typing errors the user can type in 'DISPLAY' and the system will print out all the data just entered. If an error is found, it can be corrected in the following way:

A. Correct misspelt author's name

a. erase the whole I th author's name and replace the

correct one by entering:

ATH [I;] ← AL15 ↑ 'correct name'

Example:

If the third author 'Walker, D.' was misspelt as 'WKER, D.', then it must be corrected by entering: ATH [3;] ← AL15 ↑ 'WALKER, D.'

b. correct the misspelt letter(s) by entering: ATH [I;x] ← 's'. where x can be a vector or a scalar. The value(s) of x is the position(s) of missplet letter(s). Parameter s, having the same length as x, contains letter(s).

Example:

If the second author 'WALKER, D.' was misspelt as 'WAKLER, D.', then it must be corrected by entering: ATH [2; 3 4] ← 'LK'

B. The methods for correcting the text information, descriptor, and owner are the same as for correcting a author's name except different parameters are used. The formats are as follows:

Text information	TEX ←'the correct text information'
	TEX [x] ← 's'
Descriptor	KEY [I;] ← L45 ↑ 'correct descriptor'
	KEY [I;x] ← 's'
Owner	OWNER [I;] ← L15 ↑ 'correct name'
	OWNER [I;x] ← 's'

C. If serious error is found, the user can just ignore it by entering 'ENTER' to start again.

4. Data storage

After data for one document has been entered the user must store it by executing the function STORE or COLLECT .

enter: STORE

or

enter:)COPY FNS COLLECT

COLLECT

5. Document retrieval (see Figure III-7 for algoirthm)

When the user starts the search request, he should access the file, copy the retrieval program function over, then do the search request.

A. Retrieval of document by name

a. access to the file by entering:

)LOAD Ws (s=1,3,..,2n-1)

b. the grouped function-RETRIEVE1 should be in Ws, if not, enter:)COPY FNS RETRIEVE1

c. enter: BYAUTHOR

d. enter author's name by last name, coma, blank, then initial of the first name followed by a period, or enter the specific name.

System's Action

If the entered author's name is found, the system will print out all the documents written (identified) by the name entered. If the entered name is not found, the system will print out a message for the user.

e. If there are no more search requests to be entered, depress the carriage return key.

Example:

```
    )LOAD W1    ---user
    )COPY FNS RETRIEVE1     ---user
    BYAUTHOR                ---user
ENTER NAME:                 ---system
    FLECK, A.               ---user
DOCUMENT NO.4               ---system
FLECK, A.
TOWARD A THEORY OF DATA STRUCTURES, J. COMPUTER AND
SYSTEM SCIENCE 5.(OCT. 1971), 475-488
OWNER:
FISHER
DOCUMENT NO.6
FLECK, A.
ON THE COMBINATORIAL COMPLEXITY OF CONTEXT-FREE
GRAMMARS. (IN PROC. OF IFIP CONGRESS 71, 1972)
OWNER:
FISHER
HANKLY
ENTER NAME:
    (when there are no more names to be retrieved,
     the user should depress the carriage return
     key.)
```

Algorithm for Search Request

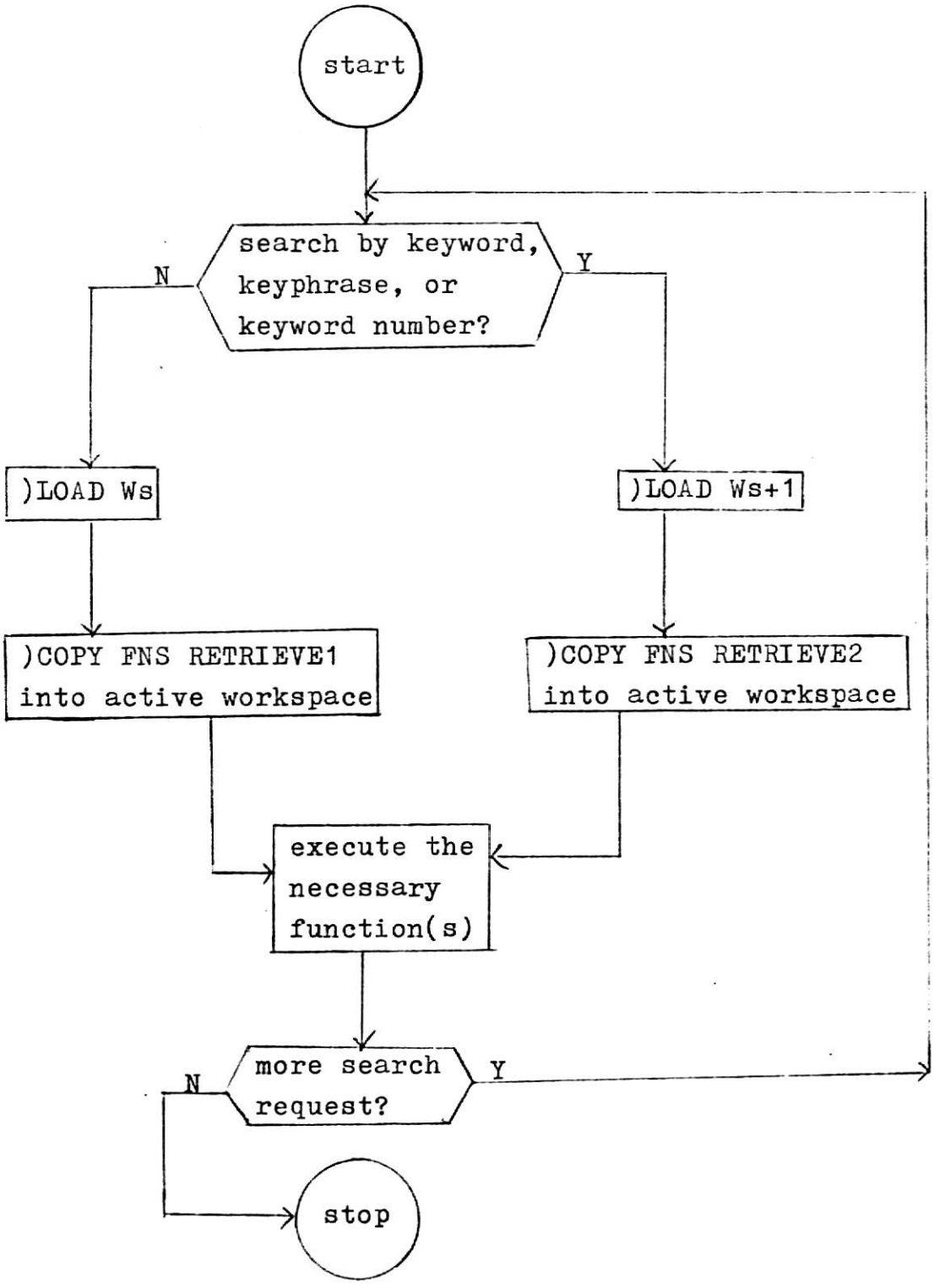


Figure III-7

If the entered name is not in the file, the system will respond: 'entered name' IS NOT IN W1 .

B. Retrieval of documents by keyword, keyphrase, or subject

User's Action

a. access to the data file by entering:
)LOAD Ws+1 (s=1,3,..,2n-1)
b. have the grouped functions—RETRIEVE2 in Ws+1, if not, enter:)COPY FNS RETRIEVE2
c. enter: BYKEYWORD
d. enter: keyword (keyphrase)
e. When finished with search request, the user should depress the carriage return key.

System Response

If the entered descriptor is found, the format as described in Figure III-8 will be printed out indirectly, otherweise the "not found" message will be printed out.

```
DOCUMENT NO.xx
author name
text information
OWNER:
owner's name
```

Figure III-8

C. Retrieval of a course reading list

User's Action

a. access to the file by entering:
)LOAD Ws (s=1,3,..,2n-1)
b. The grouped functions—RETRIEVE1 should be in Ws, if not, enter:)COPY FNS RETRIEVE1
c. enter:READINGS
d. enter: specified course name

System Response

a. If the search request is matched, the system prints out reading list as the format described in Figure III-8.
b. If the search request is not matched, a message will be printed out.

D. Retrieval of document by document number

User's Action

a. acess to the data file by entering:
)LOAD Ws (s=1,3,..,2n-1)
b. the grouped functions—RETRIEVE1 should be in Ws, if not, enter:)COPY FNS RETRIEVE1
c. enter: BYBOOKNUMBER
d. enter one or more document numbers at a time

System Response

a. prints out the information as the format described in Figure III-8

b. If the entered document number is not in the file an execution error message will be printed out.

E. Retrieval of documents by descriptor(keyword or key-phrase) number

User's Action

a. access to the data file by entering:
)LOAD Ws+1 (s=1,3,..,2n-1)
b. the grouped functions—RETRIEVE2 should be in Ws+1, if not, enter:)COPY FNS RETRIEVE2
c. enter: BYKEYNUMBER
d. enter one or more keyword numbers at at time

System Response

a. prints out the document information as the format described in Figure III-8, indirectly
b. if the entered number is not in the file, no information will be printed out.

F. Other document retrieval method

Each Ws/Ws+1 has a group named DATA1/DATA2 which contains the data bases for document retrieval only. Instead of entering ")LOAD Ws(Ws+1) ", the user can copy DATA1(DATA2) into his active workspace from his or other user's library.

6. Document deletion

User's Action

a. in workspace Ws, if not, enter:

)LOAD Ws (s=1,3,..,2n-1)

b. enter:)COPY FNS DELETE

c. enter: DELETE

d. enter to be deleted document number

System Response

a. prints out: 'THE DOCUMENT NO. X HAS BEEN SUCCESSFULLY DELETED'

7. Physical file maintenance

It is necessary to print out all the file in the sequence of authors, of document numbers, and of descriptors as a reference for the user.

A. Prints out authors' file in alphabetical order

a. in workspace Ws, if not, enter:)LOAD Ws

b. enter:)COPY FNS PTA

c. enter: ORDAUTHS

B. Prints out file in the sequence of document numbers

a. in workspace Ws, if not, enter:)LOAD Ws

b. enter:)COPY FNS DUMP1

c. enter: SQNC

C. Prints out descriptors' data file in alphabetical order

a. in workspace Ws+1, if not, enter:)LOAD Ws+1

b. enter:)COPY FNS DUMP2

c. enter: ORDKWS

Algorithm for Deleting a Document

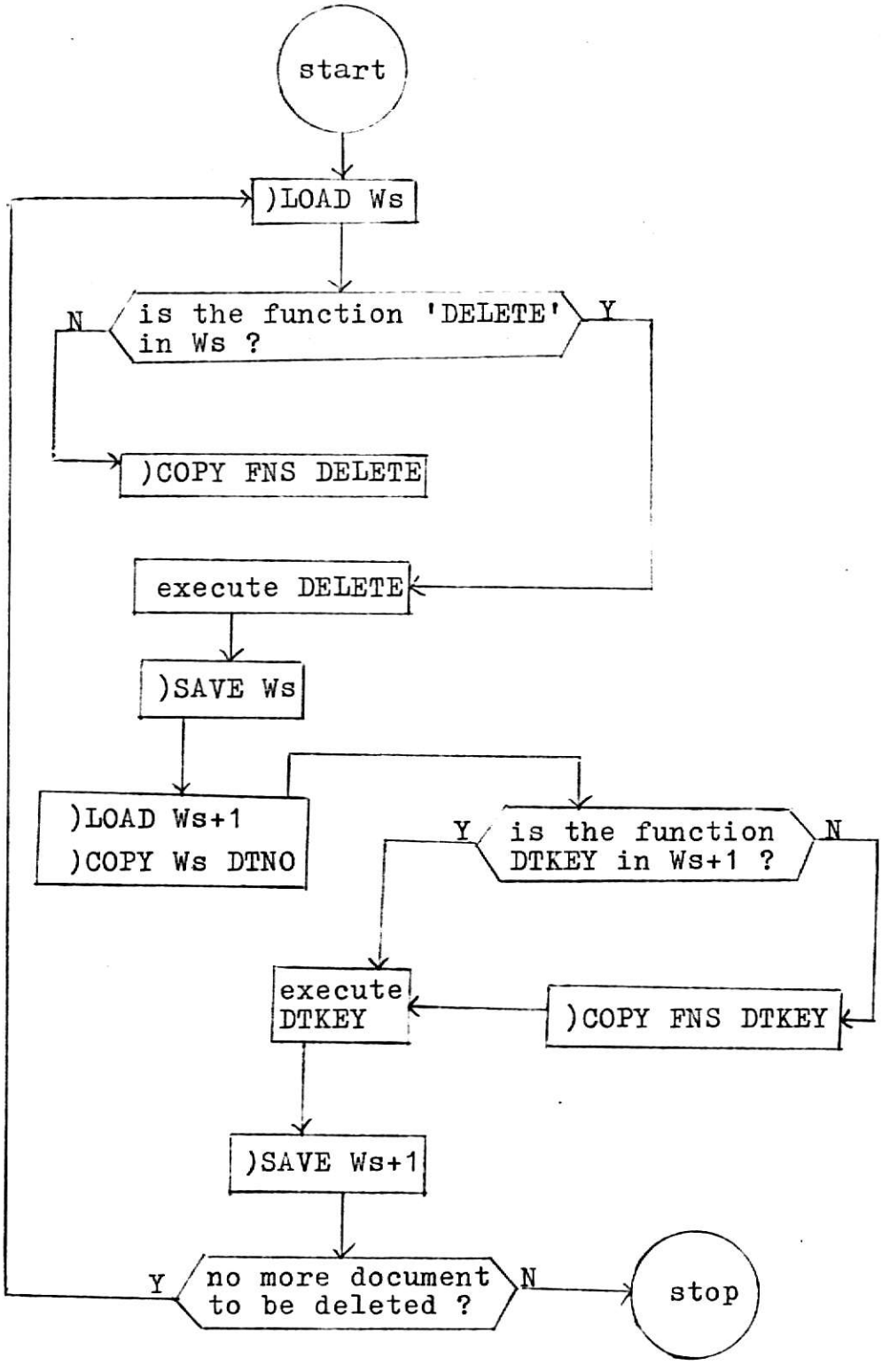


Figure III-9

8. File initialization (see Figure III-10)

The user can create as many files as he needs. If the user created n files, he may classify all the documents into n/m clsddifications (m is the number of classifications). A file should be created before a user can load the file for storage operations. The steps of creating a file are as follow:

a. enter:)LOAD FNS

b. enter: SYSGEN

c. enter:)SAVE FNS

d. follow the system instructions

Algorithm for file initialization

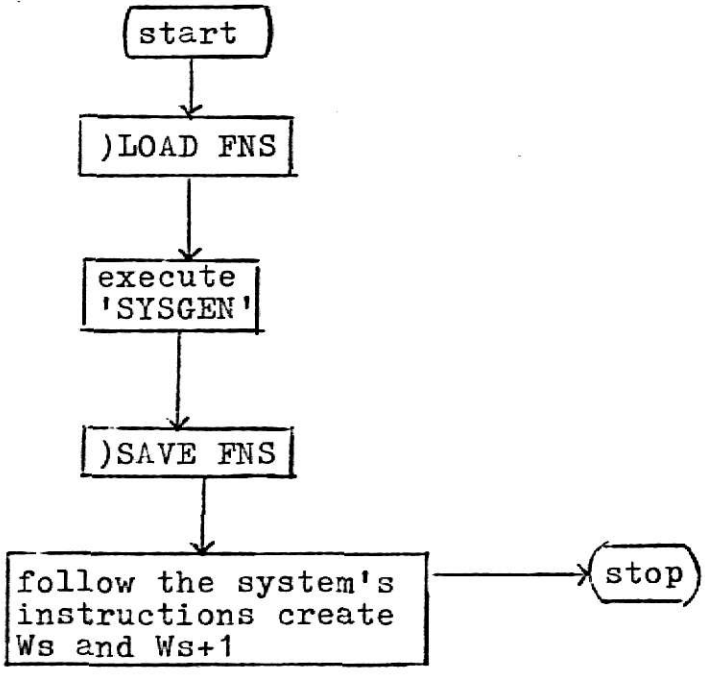


Figure III-10

III.5 Procedure Summary

1. System initialization
   a. enter:)LOAD 116087 FUNCTIONS (116087 is library number)
   b. enter:)WSID FNS
   c. enter: SYSGEN
   d. enter:)SAVE FNS
   c. follow the instructions given by the system
2. Data entry
   A. Enter a document
      a. enter:)LOAD Ws (s=1,3,..,2n-1)
      b. enter:)COPY FNS CREATION1 (as needed)
      c. enter: ENTER
         A_followed by a name(author or identifier, if no name, enter: A_)
         T_followed by a line of text information
         K_followed by descriptor (if no descriptor, enter: K_)
         O_followed by owner's name
         END
      d. follow the instructions returned by the system
   B. Add a new owner's name into the system
      Enter the following instructions as needed:
)LOAD Ws
)COPY FNS NEWOWNER
      NEWOWNER
)SAVE Ws

C. Store the reading list for a course

Enter the following instructions as needed:

```
)LOAD Ws
)COPY FNS ADREADLIST
ADREADLIST
)ERASE ADREADLIST
)SAVE Ws
```

D. Add a new course name to a file

Enter the following instructions as needed:

```
)LOAD Ws
)COPY FNS NEWCOURSE
NEWCOURSE
)ERASE NEWCOURSE
)SAVE Ws
```

3. Data display and editing

a. enter: DISPLAY

b. check errors

c. corrected by the following way:

```
ATH [I;] ← AL15 ↑ 'name'
ATH [I;x] ← 's'

TEX ← 'correct text'
TEX [x] ← 's'

KEY [I;] ← L45↑ 'correct keyword'
KEY [I;x] ← 's'

OWNER [ I; ] ← L15 ↑ 'correct name'
```

OWNER [I;x] ← 's'

If a serious error is found, type in :ENTER and start again.

4. Data storage

A. enter:STORE

or

B. enter:

)COPY FNS COLLECT (as needed)

COLLECT

5. Document retrieval

A. by name

enter the following instructions as needed:

)LOAD Ws (s=1,3,..,2n-1)

)COPY FNS RETRIEVE1

BYAUTHOR

When finished with the search, depress the carriage return key.

B. by document number(s)

enter the following instructions as needed:

)LOAD Ws (s=1,3,..,2n-1)

)COPY FNS RETRIEVE1

BYBOOKNUMBER

C. by keyword, key phrase, subject, or descriptor

enter the following instructions as needed:

)LOAD Ws+1 (s=1,3,..,2n-1)

)COPY FNS RETRIEVE2

BYKEYWORD

When finished with the search, depress the carriage return key.

D. by keyword, key phrase, subject, or descriptor number(s)

enter the following instructions as needed:

)LOAD Ws+1 (s=1,3,..,2n-1)

)COPY FNS RETRIEVE2

BYKEYNUMBER

E. search course reading list

enter the following instructions as needed:

)LOAD Ws (s=1,3,..,2n-1)

)COPY FNS RETRIEVE1

READINGS

F. other method

instead of loading a workspace, copy DATA1(or DATA2) from Ws(or Ws+1)

6. Document deletion

enter the following instructions as needed:

)LOAD Ws (s=1,3,..,2n-1)

)COPY FNS DELETE

DELETE

7. Physical file maintenance

   A. author (name of document) data file in alphabetical order

      a. in workspace Ws, if not, enter:)LOAD Ws

      b.)COPY FNS PTA

      c. ORDAUTHS

   B. in the sequence of document numbers

      a. in workspace Ws, if not, enter:)LOAD Ws

      b.)COPY FNS DUMP1

      c. SQNC

   C. descriptor(keyword, key phrase) data file in alphabetical order

      a. in workspace Ws+1, if not, enter:)LOAD Ws+1

      b.)COPY FNS DUMP2

      c. ORDKWS

8. File initialization

   enter:

)LOAD FNS

   SYSGEN

)SAVE FNS

Appendix I

DETAILS OF DATA STRUCTURES USING APL

A general description of ILCRS file organization using APL is described by Figure A-I-5. All the details are:

1. Structure of title, publisher, vol. No., page, date, bibliographic reference information, and/or any other information

   The above information is stored in the records named TEXTS. TEXTS is structured as contiguous sequential records. Each record contains N blocks, each block has a length of L50 characters (L50 is a parameter defined in system initialization procedure.), that is each text information is divided into N lines, each line has L50 characters. The total number of records equals to the total number of documents stored in the system (see Figure A-I-1).

   TEXTINDX— is used to index into TEXTS of last character of i th document information. The value of each element is equal to L50 times X (L50 is a parameter), where X is an integer parameter. TEXTINDX is a vector which contains DNO - 1 elements, where DNO - 1 is the current number of documents stored (see Figure A-I-1).

Text Structures

TEXTS

```
SYSTEM ANALYSIS FOR DATA TRANSMISSION. PRENTICE-HALL,                  65
ENGLEWOOD CLIFFS, N. J., 1972, 909 PP.                                 130
COMPUTERIZED STORAGE AND DIALOG-RETRIEVAL OF PHILOSOPHICAL             195
INFORMATION. COMPUTER AND THE HUMANITIES, VOL 5,NO.5,4/74              260
BALLOTS-BIBLIOGRAPHIC AUTOMATION OF A LARGE LIBRARY USING A            325
TIME-SHARING SYSTEM. COMPUTERS AND HUMANITIES,VOL 5,NO.5,71            390
SOME INHERENT AMBIGUITY OF CERTAIN CONTEXT-FREE LANGUAGES.             455
DEP. OF MATH.,HEBREW U,JERUSALEM,ISRAEL,INFOR & CONTROL,VOL.           520
18, NO. 4, MAY 1971, PP. 355-362                                       585
--- - ----  -- ---- -- ------ -- - ----  --- ----- ---- --
---- ---   -- - --   -----   --- ---- ------   ----  --- -

TEXTINDX   130 260 390 585  --- --- ----
```

Figure A-I-1

2. Structure of author, and author index records (see Figure A-I-2)

The author's names with the same initial letter are stored in close proximity to each other. The length of each name is AL15 (AL15 is a parameter defined in system initialization procedure) characters by cutting off the tail part and keeping the first AL15 characters. The names are recorded, last name first, followed by the initial of the first name (if publisher or organization is used, they are recorded by standard abbreviation).

A list of names are stored in the array--$AUTHS_{NA \times AL15}$, where NA is initial total number of names to be in the file.

Each row of index of authors' books array (AUTHBKS) is one to one correspondence to the list of authors array (AUTHS).

AUTHBKS-- is an array of NA by BL3 initially, where BL3 can be automatically expanded to the requested size. Each name is allowed to have BL3 documents, initially. If it is necessary to have up to more than BL3 documents, the AUTHBKS array will be expanded.

AUTHINDX-- is a 28-vector of first letter indices into the next available locations of AUTHS.

LETTERS -- is a 27-vector of letters (the contents are

' ABCDEFGHIJKLMNOPQRSTUVWXYZ'

STARTINDX-- the starting storage position of each letter

The initial assignment for the size of storage spaces of each letter, the contents of AUTHINDX, depends on the expected number of occurrences of each letter in AUTHS. In other words, the number of persons whose last names begin with the initial letter A may be less than those with B. Under such a condition, we may assign more storage spaces for B than for A. As to the expected value for each letter, this depends on the user's estimation (see program--DEFAULT for example).

SPACEINDX -- the size of each bucket for the storage locations of each letter

3. Documents and authors corss reference structure

BKAUTHS -- is a NB by AL3 two dimensional matrix. The content of BKAUTHS [I;J] is x, where x stands for i the document written by x th author whose name was stored in x th row of AUTHS (see Figure A-I-4).

4. Owners and index of document owner structures

OWNERS -- is a list of owners, with NWS by L15. The maximum length of owner's name is L15 characters. The initial value of total number of owners is NWS.

BKOWNERS -- is a NB by OL3 two dimensional array, where NB

is the number of documents to be in the system, OL3 means each document can have at most OL3 owners. The value of BKOWNERS [I;J] is an index number of author (see Figure A-I-4).

5. Structures for a course reading list

COURSES -- is a list of course names. It is NC by L5 two dimensional array, where NC is the initial total number of courses to be used for retrieval of reading list. The name of each course is L5 characters in length, for example, CS650, CS400,--,etc.

READLIST -- is NC by NR two dimensional array. The contents of READLIST I; is a list of reading list document numbers for i th course, where NR is the maximum number of readings for each course.

6. The structures of descriptors and descriptor index file

The keywords indexing permits the users to select several (at most MKB) descriptors for an index to describe a particular document.

KYS -- The keywords (descriptors) with the same initial letter are stored in close proximity to each other in KYS_{NKB} by L45 , where NKB is the initial total number of descriptors in a file, and L45 is the length of a descriptor (see Figure A-I-3).

KYINDX -- is a 28-vector, where each element is used to tell the next available location of each different initial letter for storing the entering descriptor(see Figure A-I-3).

KYBKS -- is a NKB by MBK two dimensional array. It is used as a cross reference for descriptor number and document number(s). For example, the content of KYBKS [I;] is a vector of document numbers which can be recognized by the i th descriptor (see Figure A-I-3).

BKYNO -- is a NB by MKB matrix, where NB is the number of documents in a file, MKB represents for each document may have up to MKB keywords (descriptors). The value of BKYNO I; is a vector for the index of i th document keywords (see Figure A-I-4).

KSTARTINDX -- the starting positions of alphabet locations for KYS.

7. Structures for deleting a document

If a document is out of date, useless, or lost, then it can be deleted from the file. The following vectors are necessary for deleting a document and storing a new document text in the deleted document location.

DNO -- to be deleted document number

TXWAIT -- is a variable length vector which keeps the available document numbers.

TXLENGTH-- keeps the avialable storage length of each deleted document number. Each element of TXLENGTH is one to one correspondent to that of TXWAIT.

Structures of Names

		AUTHS$_{NA \times L15}$	AUTHBKS$_{NA \times MAX}$	
1				
2				2=STARTINDX[1]
3		AHO, A	9 10	
4	A			4=AUTHINDX [1]
5				5=STARTINDX[2]
6				6=AUTHINDX [2]
7				10=SPACEINDX[2]
8	B			
9				
10				
11				
12				
13				
14				
15				
16				
17				
18				
19	C			
20				
21				
22				
23				
24		DIEMER, A.	3	3 is document number
25	D			
26				
27				
28				
29	E	ELLIS, C.	8	

FIGURE-A-I-2

Structures of Keywords

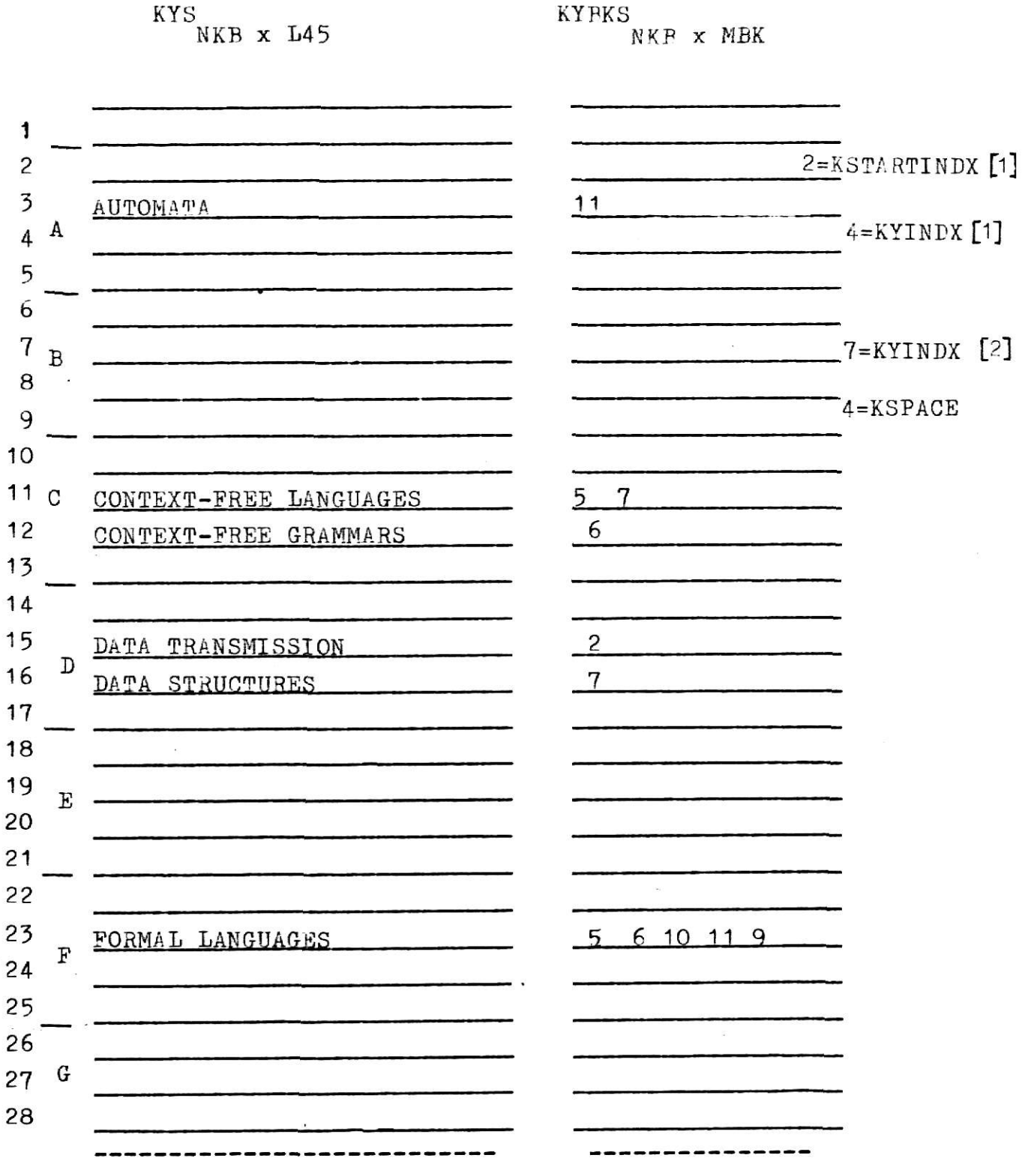


FIGURE-A-I-3

Structures of Surrogate Files

document number	BKAUTHS NB x AL3	BKOWNERS NB x OL3	BKYNO NB x MKB
1			
2	66	4	79 15
3	24 41	1 2	36 35 75
4	113	5	36 47
5	95	2	11 23
6	32	3	12 36
7	32	4	55 11
8	29	1	65 63
9	3 25	1	23 51 95
10	3 111	1 2	23 64
11	42 111	1	3 23
12			
13			
14			
15			
16			
	author number	owner number	keywords number

FIGURE-A-I-4

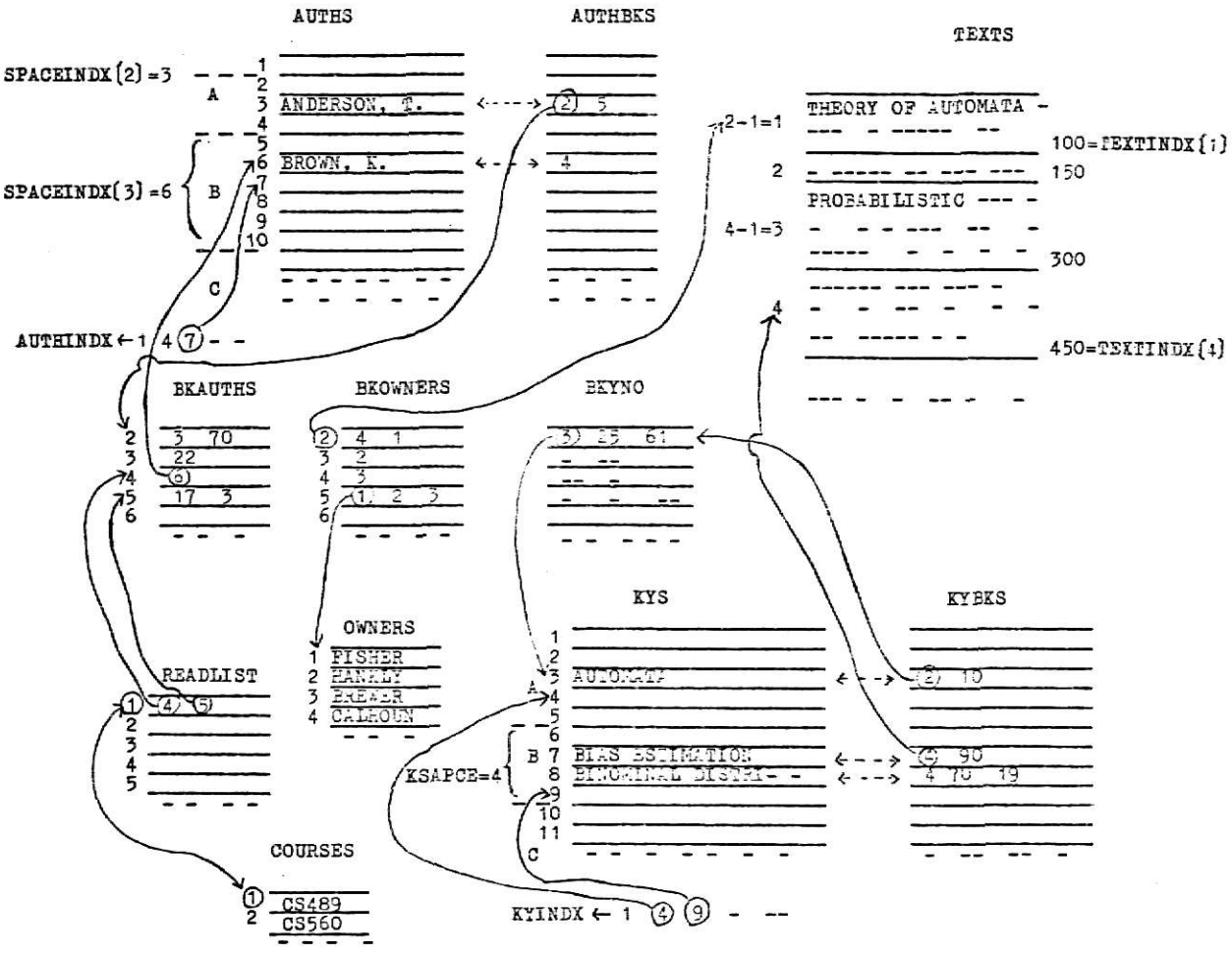

AUTHS
AUTHBKS
TEXTS
SPACEINDX[2]=3
SPACEINDX[3]=6
ANDERSON, T.
BROWN, K.
THEORY OF AUTOMATA
PROBABILISTIC
100=TEXTINDX[1]
150
300
450=TEXTINDX[4]
AUTHINDX
BKAUTHS
BKOWNERS
BKYNO
OWNERS
FISHER
HANLEY
BREAUX
CALHOUN
READLIST
KYS
KYBKS
AUTOMATA
BIAS ESTIMATION
KSAPCE=4
COURSES
CS489
CS560
KYINDX

Figure-A-I-5

Summary of Data Structures

variable	description	current size (value)	initial size (value)
AL15	the length of name		
ATH	temporary storage location for names	A4 x AL15	"
AUTHS	list of authors' names	NA x AL15	"
AUTHBKS	index of author's books array	NA x MAX	NA x BL3
AUTHINDX	a 28-vector of first letter indicex into AUTHS	28	"
BKAUTHS	index of document authors	NB x AL3	"
BKOWNERS	index of document owners	NB x OL3	"
BKYNO	index of document descriptors(keywords)	NB x MKB	"
COURSES	list of course names	NC x L5	"
DNO	current document number		(1)
DTNO	to be deleted document number		
KEY	temporary storage location for a document keywords	MKB x L45	"
KSPACE	storage bucket size of each letter		
KSTARTINDX	the starting position of each letter in KYS	28	"
KYBKS	index of keyword document array	variable	NKB x MKB
KYS	list of keywords	variable	NKB x L45
L45	length of descriptor		
L5	length of course name		
LETTERS	27-vector of letters	27	"

MKB	maximum number of descriptors for a document		
NA	total number of names	(variable)	
NB	total number of books in a file		
NC	total number of courses	(variable)	
NKB	initial number of keywords in a file		
NR	number of reading list for each course		
NWS	number of owners	(variable)	
OWNERS	a list of owners' names	NWS x L15	"
OWNER	temporary storage location for a document owners	OL3 x L15	"
READLIST	document index of i th course	NC x NR	"
STARTINDX	the starting position of each letter in AUTHS	28	"
TEX	temporary storage location for a document text	variable	
TEXTS	a vector for storing documents' texts	variable	
TXWAIT	waiting document numbers	variable	
TXLENGTH	available storage sizes of deleted documents	same as TXWAIT	

Appendix II

ORGANIZATION OF COMPUTER PROGRAMS

The total number of Function Programs is 28. The structures of function call can be seen in Figure III-2, Figure III-3, and Figure III-4.

A. Definition of each Function Program

1. ADATH -- store the the names into the appropriate locations, reallocate AUTHS and AUTHBKS
2. ADKEYS -- store the keywords (descriptors) into the appropriate locations, reallocate KYS and KYBKS
3. ADOWN -- store the document owners into appropriate locations
4. ADTEX -- store the text information into the appropriate location
5. ADREADLIST -- store the reading list document numbers into the appropriate locations
6. BYAUTHOR -- retrieval of documents by name
7. BYBOOKNUMBER -- retrieval of document information by document numbers
8. BYKEYNUMBER -- retrieval of document information by keyword (descriptor) number(s)
9. BYKEYWORD -- retrieval of documents by descriptor

10. COLLECT -- to collect the available storage space from TEXTS
11. DEFAULT -- to define the size of data base by default values, or to redefine part of the default values
12. DEFINE -- define the value of each parameter
13. DELETE -- to delete author, text, and owner of a document
14. DISPLAY -- for user to check the entering data
15. DTKEY -- to delete the keywords of a document
16. ENTER -- to read in the information of a document
17. INIKFL -- to initialize keyword data base
18. INITIALIZE -- to initialize parameters which have to be generated to the definite size as defined in function 'DEFINE' or 'DEFAULT
19. NEWCOURSE -- to add a new course name into the file
20. NEWOWNER -- to add a new owner's name into the file
21. ORDAUTHS -- to print out the whole file in alphabetical order of authors' names
22. ORDKWS -- to print out keyword data base in alphabetical order
23. PRINTOUT -- to print out the requested documents information
24. READINGS -- retrieval of reading list for a course
25. SORT -- sort the index of numbers into ascending order
26. SQNC -- to print out the information of the whole file in the sequence of document numbers

27. STORE -- store the entering data into the file
28. SYSGEN -- to generate a file or the system

B. Definition of each Group

1. CREATION1 -- information input Functions, for workspace Ws
2. CREATION2 -- information input Functions, for workspace Ws+1
3. DATA1 -- data base in workspace Ws
4. DATA2 -- data base in workspace Ws+1
5. DUMP1 -- display data base, for workspace Ws
6. DUMP2 -- display data base, for workspace Ws+1
7. INIT1 -- initialization Functions, for workspace Ws
8. INIT2 -- initialization Function, for workspace Ws+1
9. KVARS -- keywords file initialization variables for both workspaces Ws and Ws+1
10. KWS -- current keywords data base for both workspaces Ws and Ws+1
11. PTA -- display authors file in alphabetical order, for workspace Ws
12. RETRIEVE1 -- retrieve-by-author, retrieve-by-document number, retrieve-by- reading list, for workspace Ws
13. RETRIEVE2 -- retrieve-by-keywords, retrieve-by-keywords numbers, for workspace Ws+1

C. Algorithms

1. algorithm for storing names (see Figure-A-II-1)
2. algorithm for storing keywords (see Figure-A-II-2)
3. algorithm for storing owner's names (see Figure-A-II-3)
4. algorithm for storing text information (see Figure-A-II-4)
5. algorithm for entering the reading list of a course (see Figure-A-II-5)
6. algorithm for search by author's name (see Figure-II-6)
7. algorithm for search by document numbers (see Figure-A-II-7)
8. algorithm for search by keywords (see Figure-A-II-8)
9. algorithm for search by keyword numbers (see Figure-A-II-9)
10. algorithm for garbage collection (see Figure-A-II-10)
11. algorithm for function DEFINE & DEFAULT (see Figure-A-II-11)
12. algorithm for function INIKFL (see Figure-A-II-12)
13. algorithm for function INITIALIZE (see Figure-A-II-13)
14. algorithm for deleting a document (see Figure-A-II-14)
15. algorithm for deleting keywords (see Figure-A-II-15)
16. algorithm for entering a document (see Figure-A-16)
17. algorithm for entering a new course name (see Figure-A-II-17)
18. algorithm for entering a new owner's name (see Figure-A-II-18)

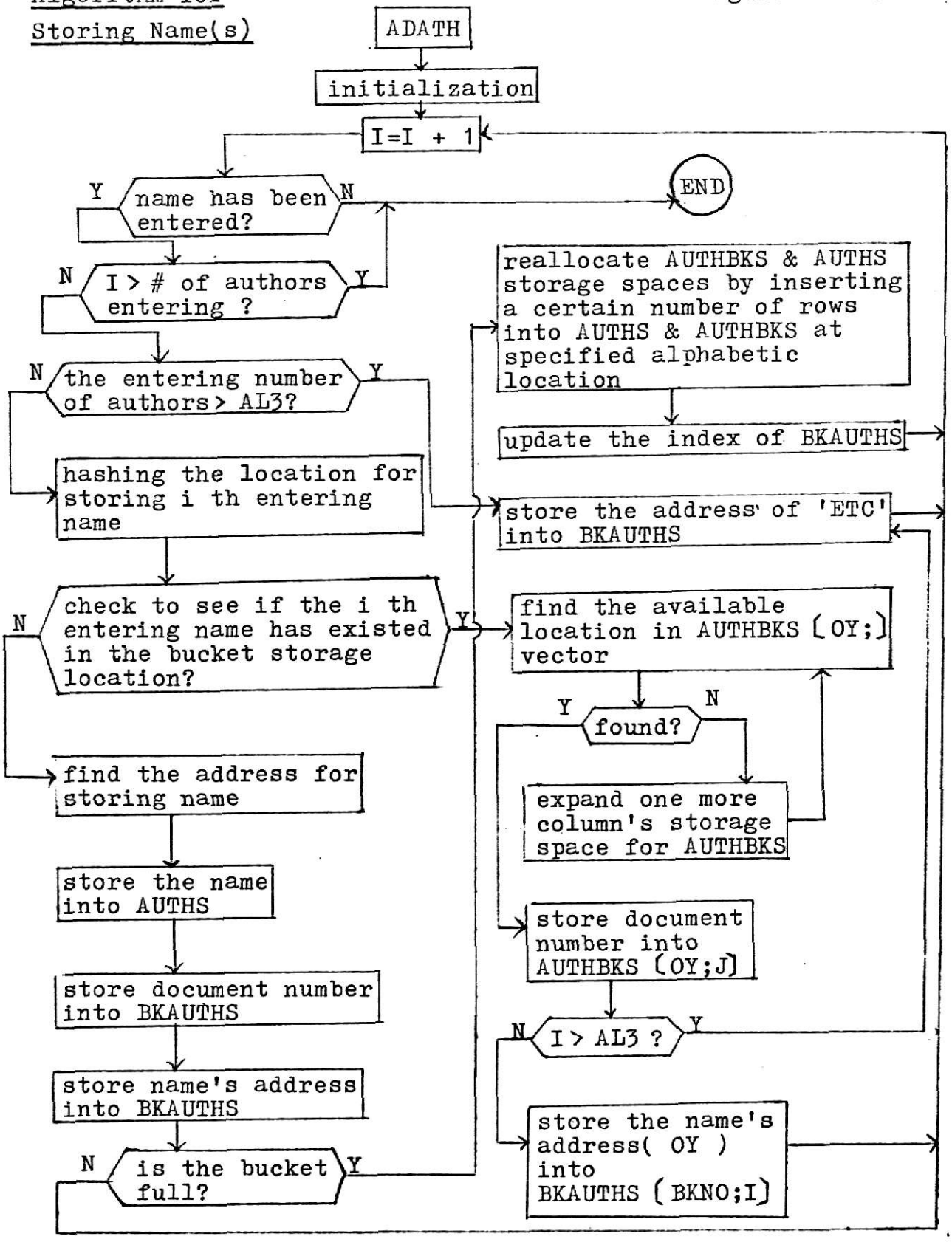
19. algorithm for printout the file in alphabetical order of names (see Figure-A-II-19)
20. algorithm for printout sorted keywords and document numbers cross references (see Figure-A-II-20)
21. algorithm for function PRINTOUT (see Figure-A-II-21)
22. algorithm for search reading list of a course (see Figure-A-II-22)
23. algorithm for sorting by index (see Figure-A-II-23)
24. algorithm for getting the sequence of document numbers (see Figure-A-II-24)
25. algorithm for function STORE (see Figure-A-II-25)
26. algorithm for function SYSGEN (see Figure-A-II-26)
27. algorithm for creating a library named FUNCTIONS (see Figure-A-II-27)
28. algorithm for system initialization (see Figure-A-II-28)

THIS BOOK CONTAINS NUMEROUS PAGES WITH DIAGRAMS THAT ARE CROOKED COMPARED TO THE REST OF THE INFORMATION ON THE PAGE.

THIS IS AS RECEIVED FROM CUSTOMER.

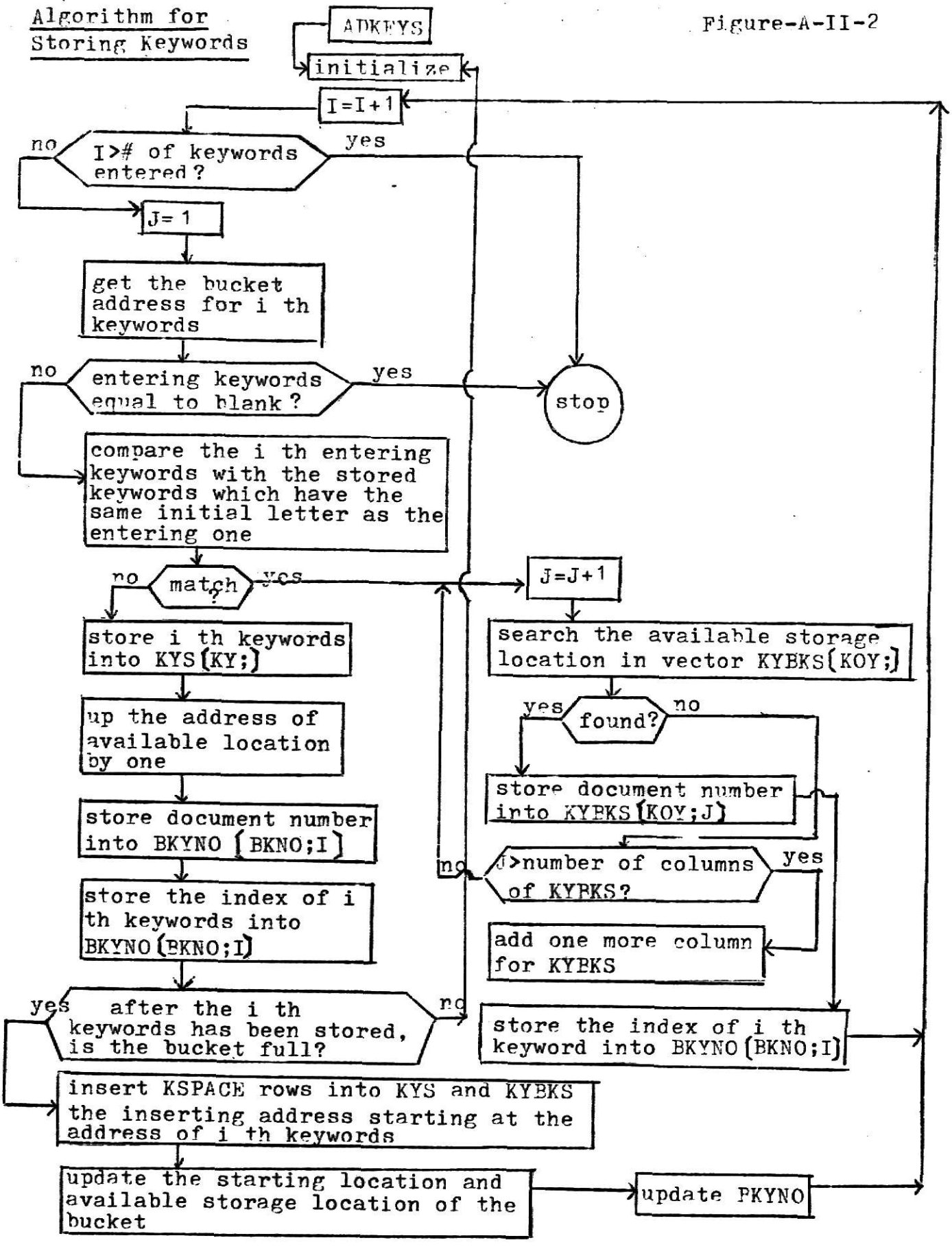
Algorithm for Storing Name(s)

Figure A-II-I



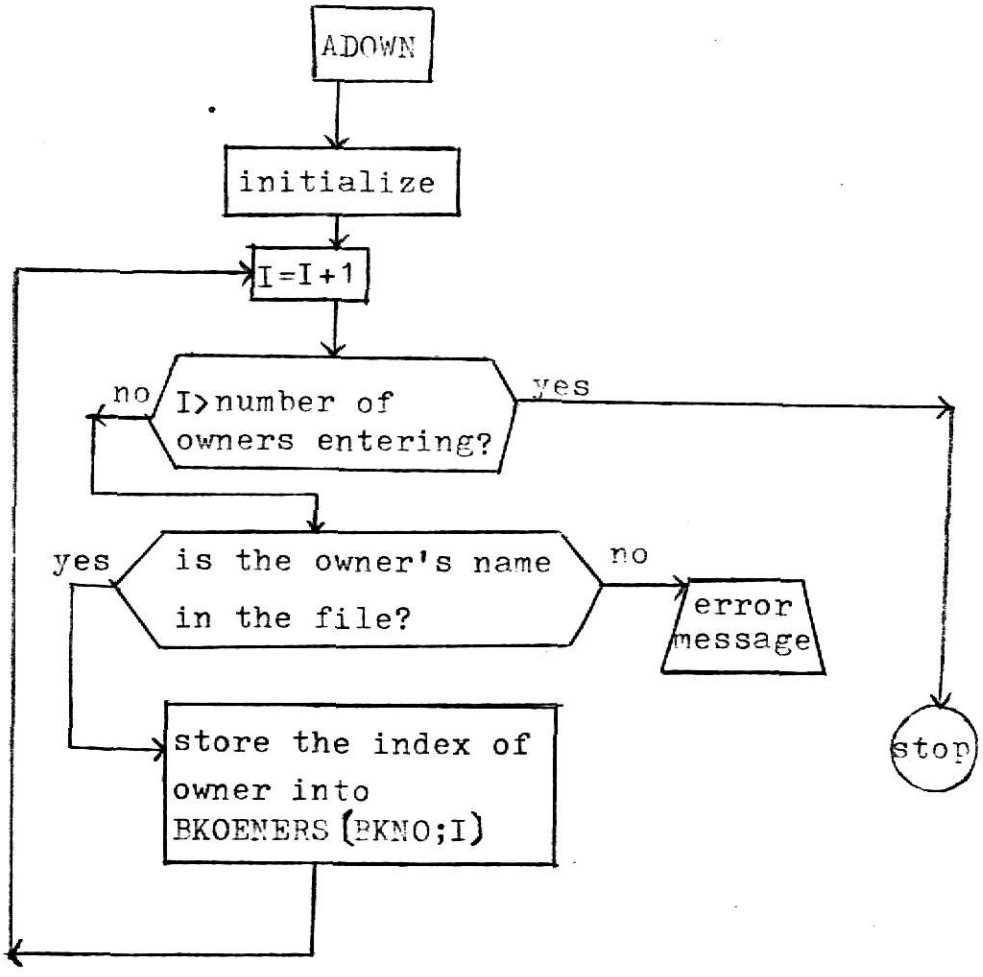
Algorithm for Storing Keywords

Figure-A-II-2



Algorithm for Storing
Owner's Name

Figure-A-II-3



Algorithm for Storing
Text Information

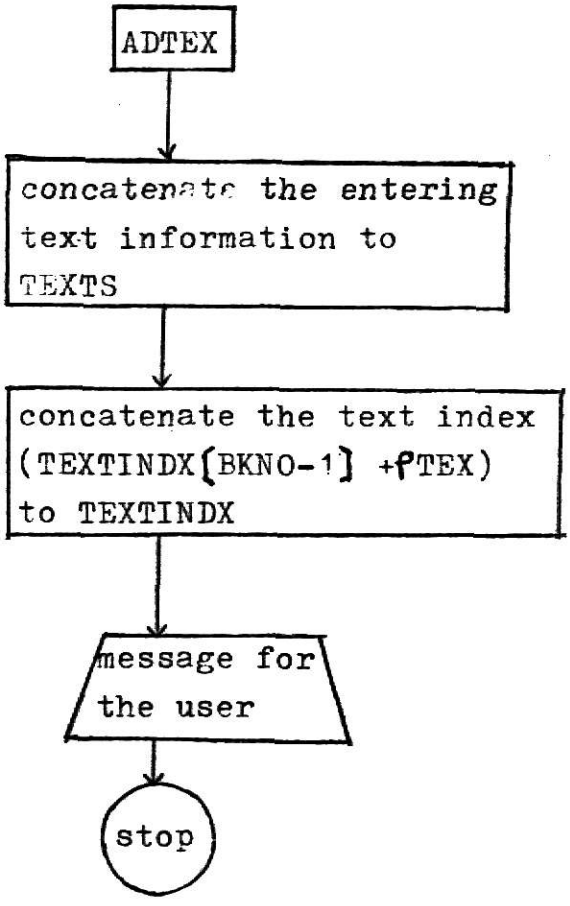


Figure-A-II-4

Algorithm for Entering the Reading List of a Course

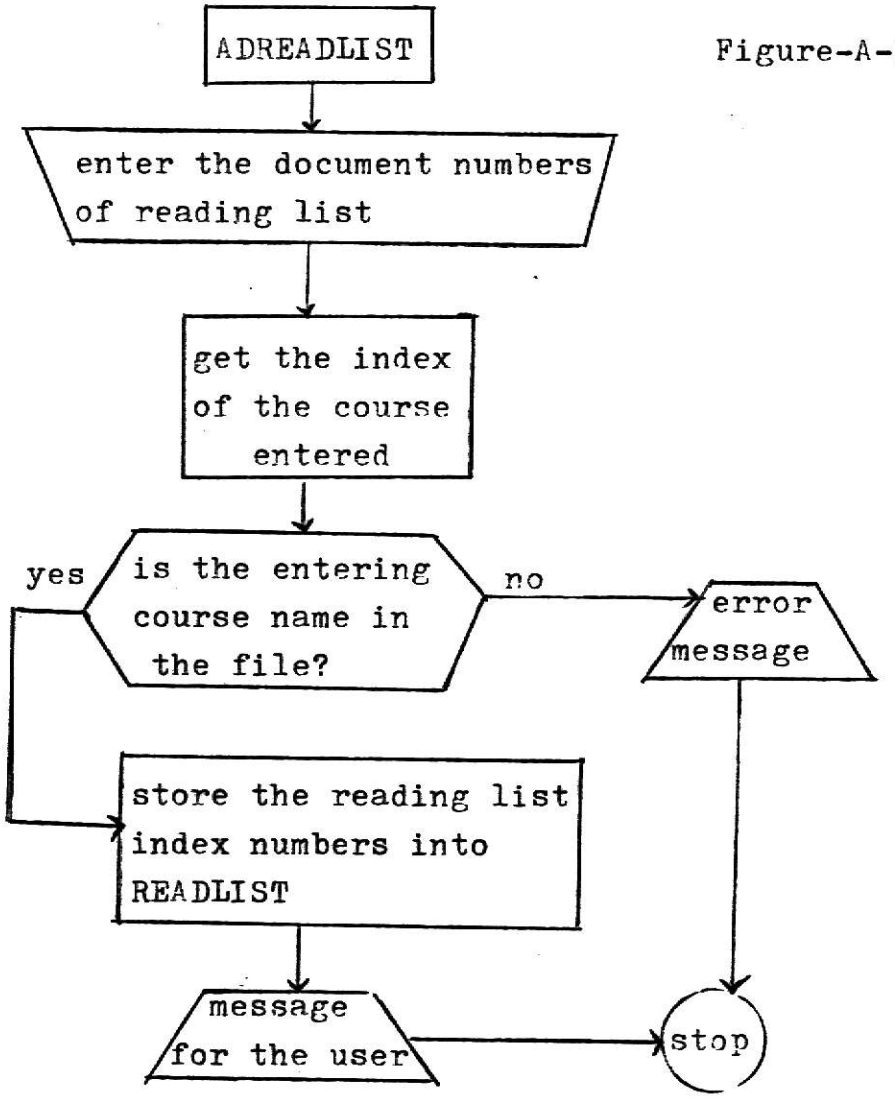


Figure-A-II-5

Algorithm for Search
by author

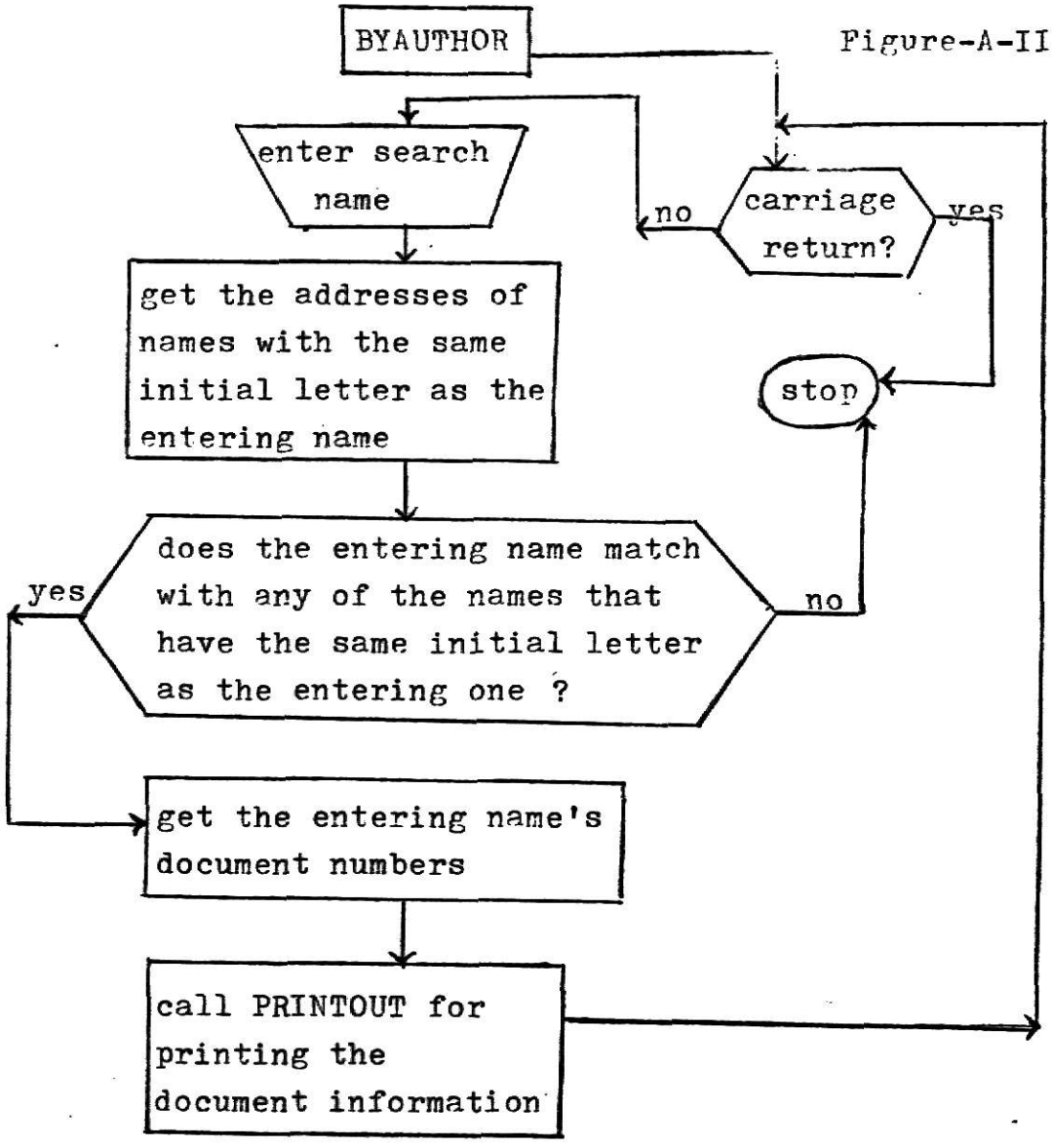
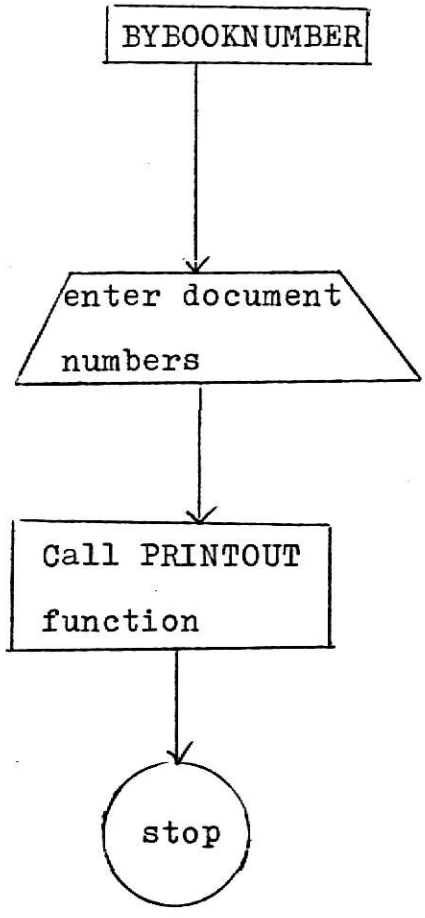


Figure-A-II-6

Algorithm for

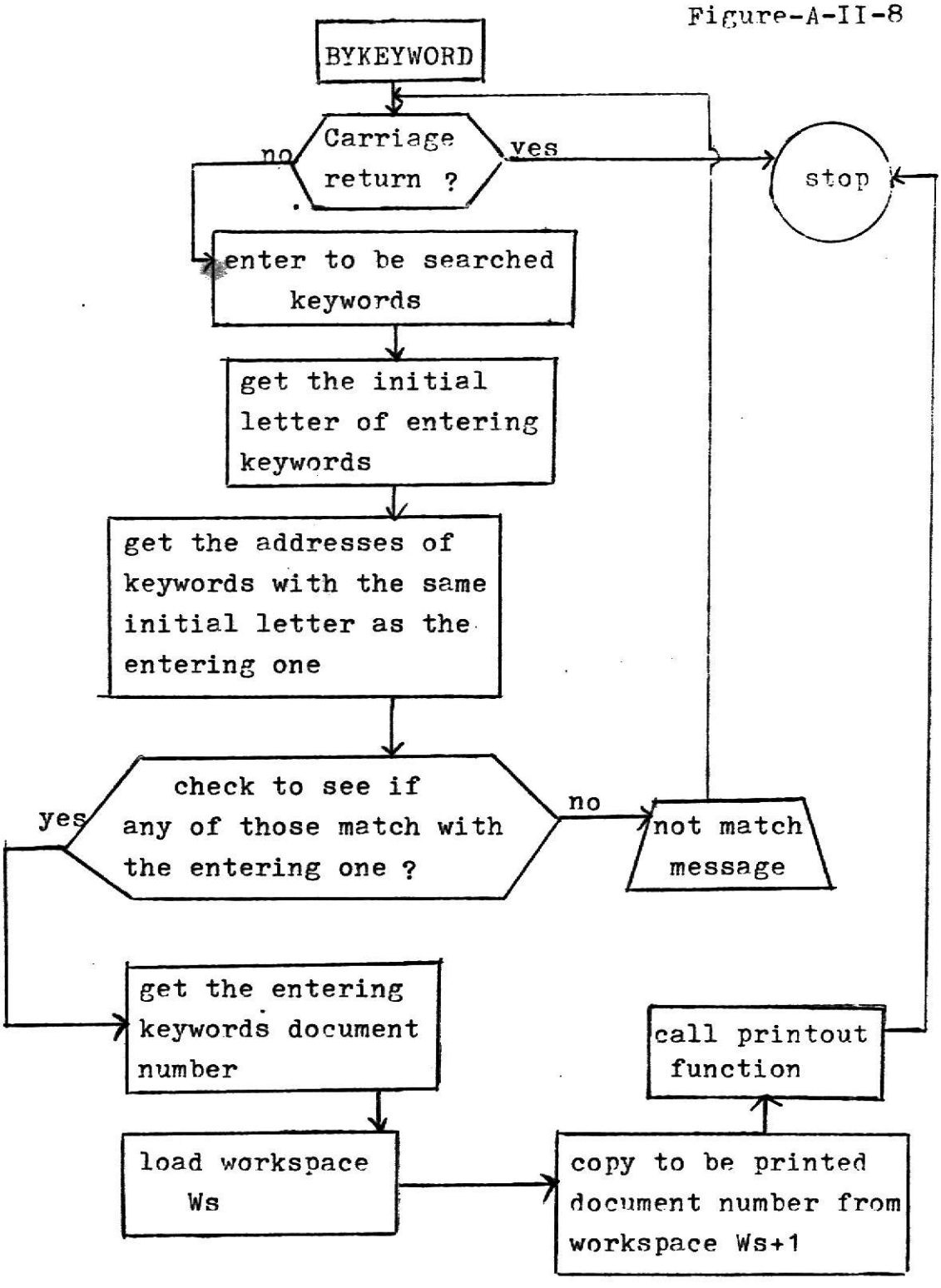
Search by Document Numbers

Figure-A-II-7



Algorithm for Search
by Keywords

Figure-A-II-8



Algorithm for Search
by Keywords Number

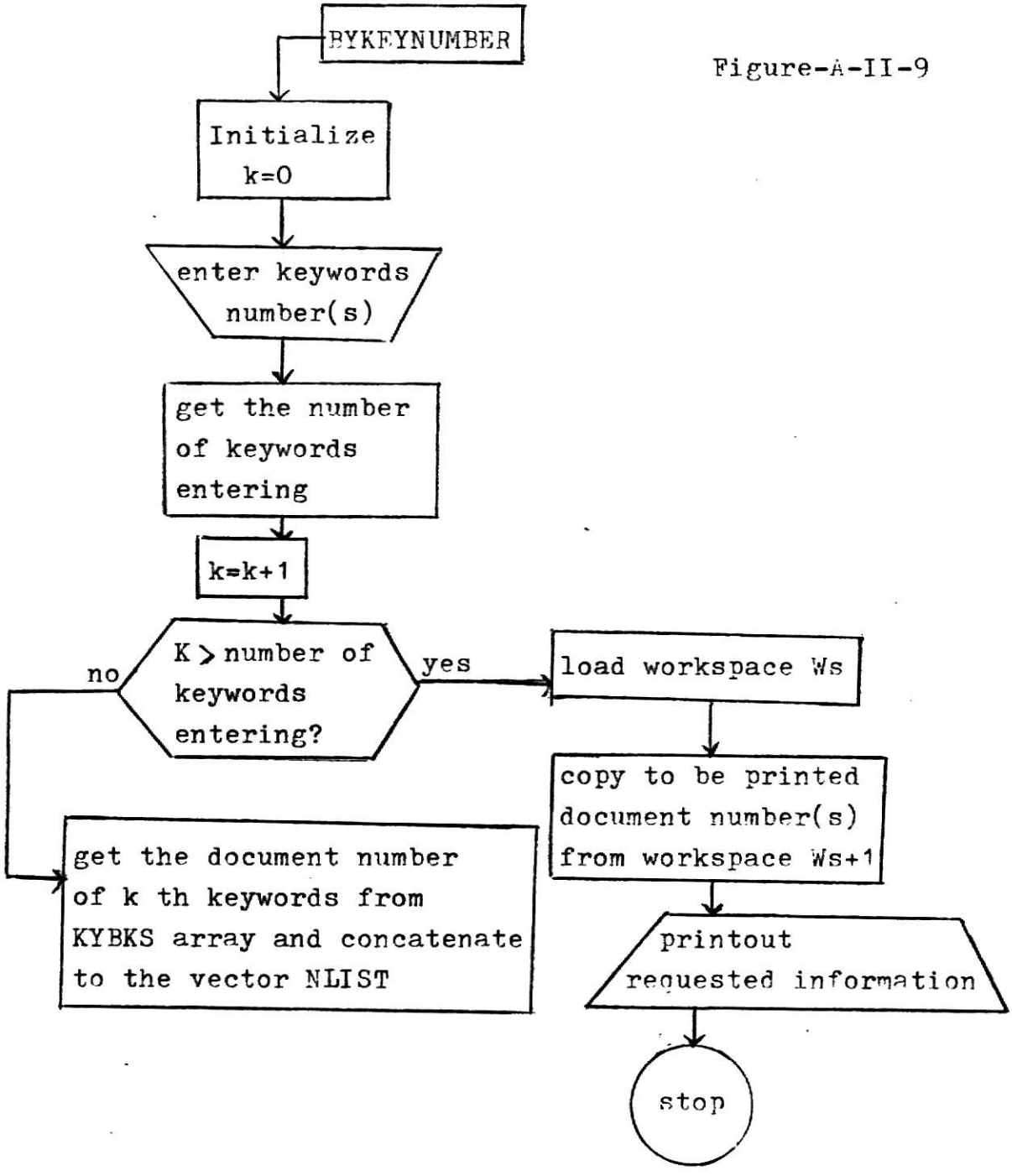
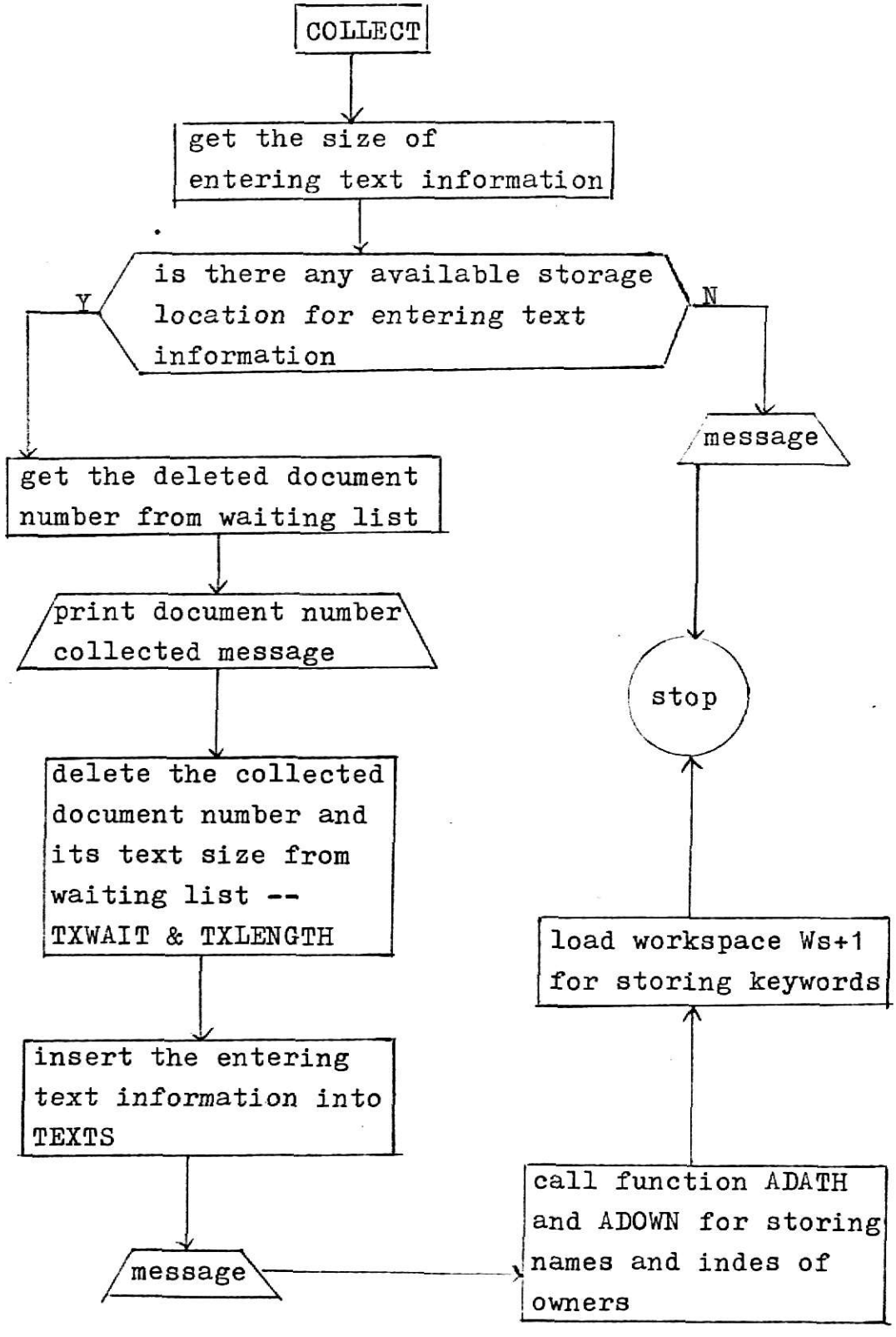


Figure-A-II-9

Algorithm for

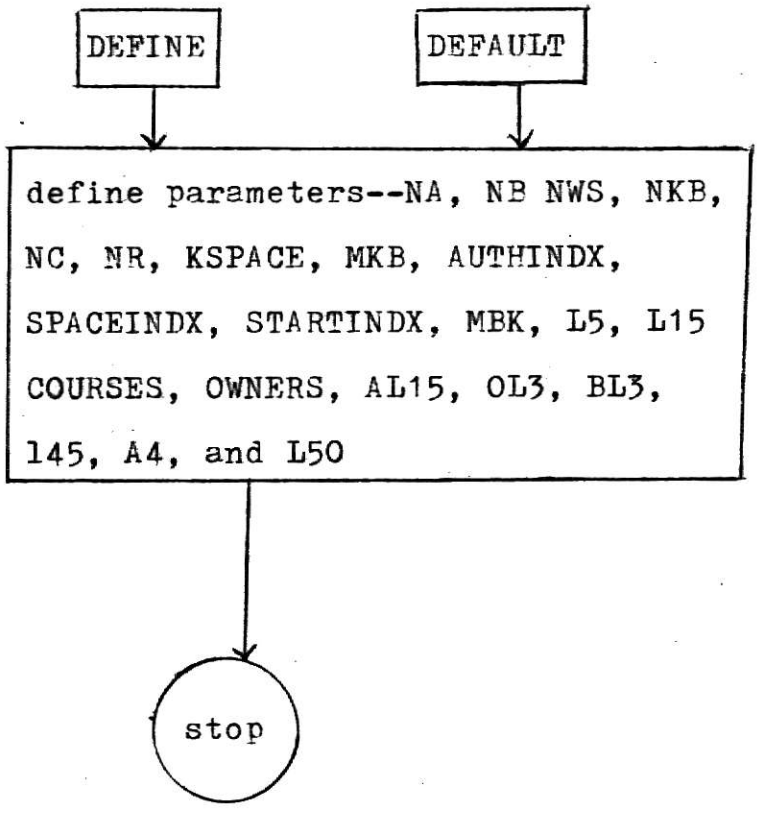
Garbage Collection

Figure-A-II-10



Algorithm for function
DEFINE & DEFAULT

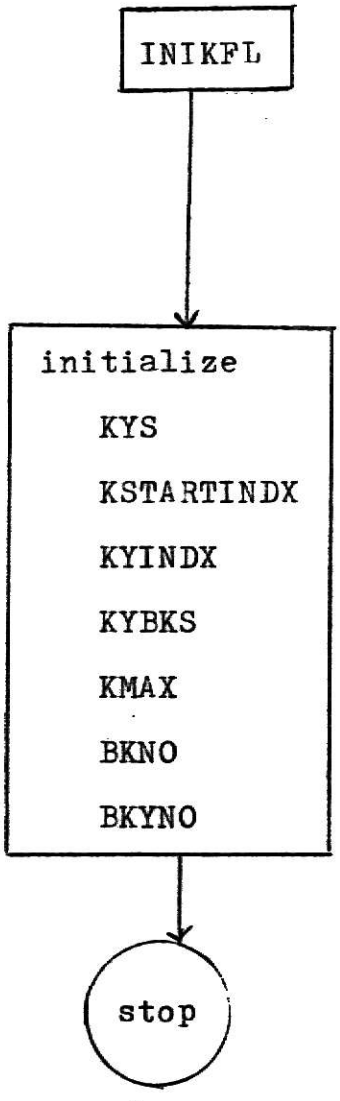
Figure-A-II-11



Algorithm for

function INIKFL

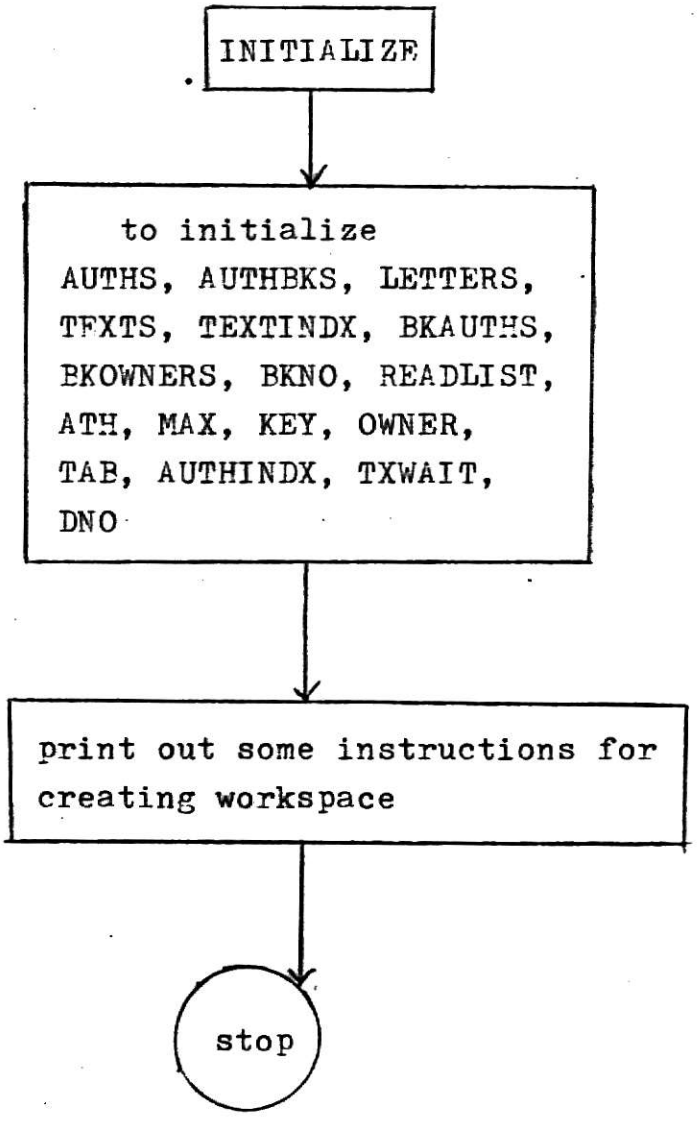
Figure-A-II-12



Algorithm for function

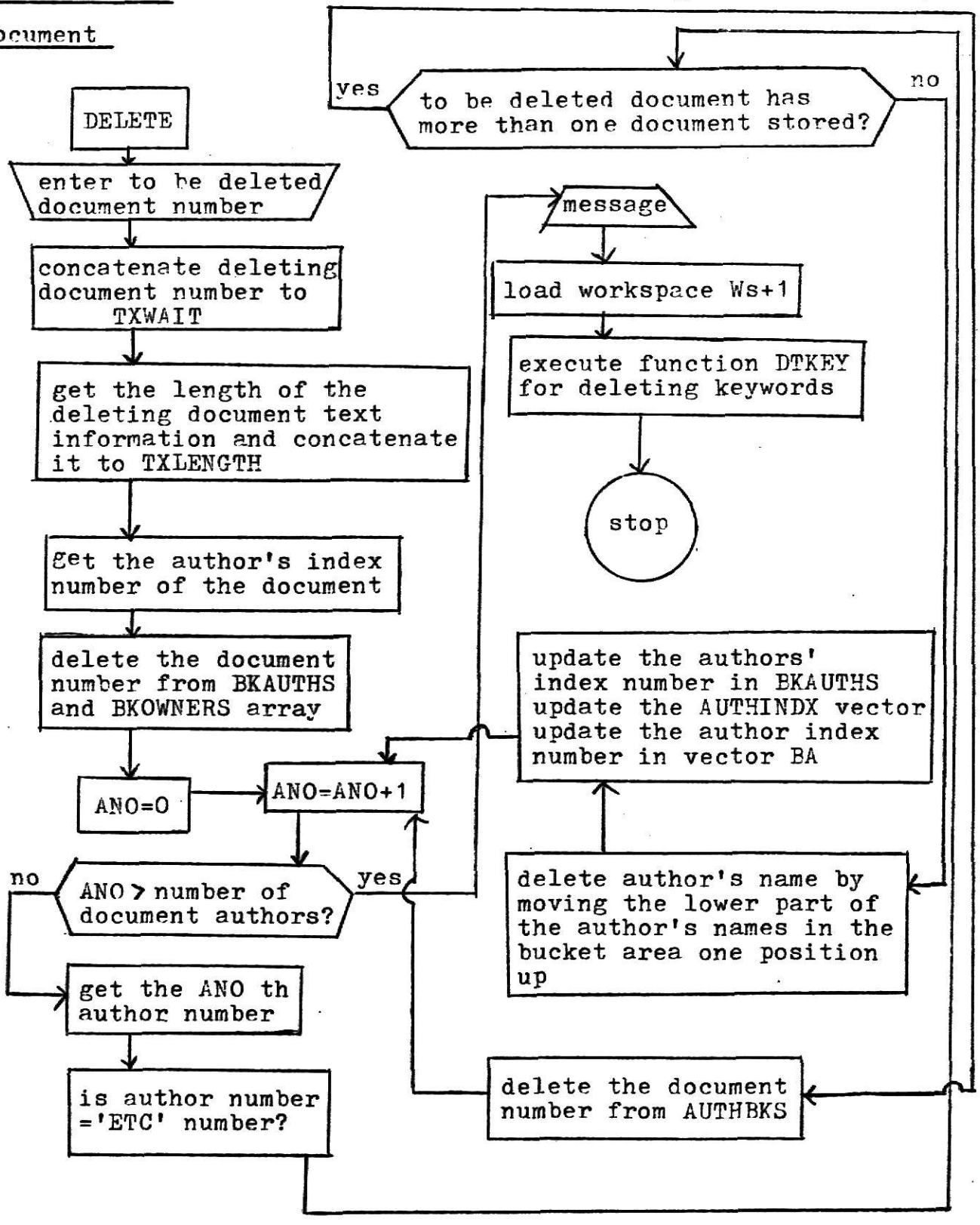
INITIALIZE

Figure-A-II-13



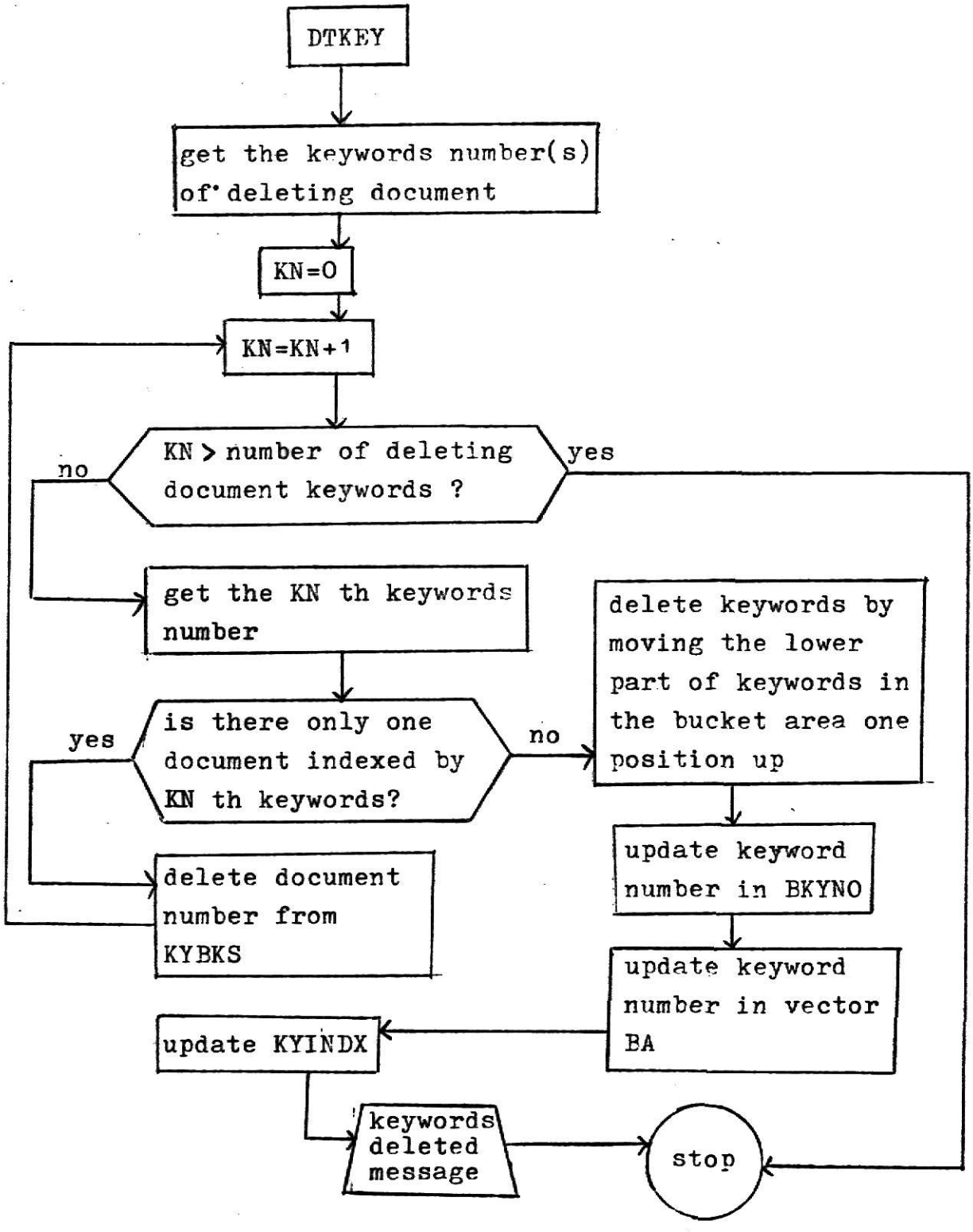
Algorithm for Deleting a Document

Figure-A-II-4



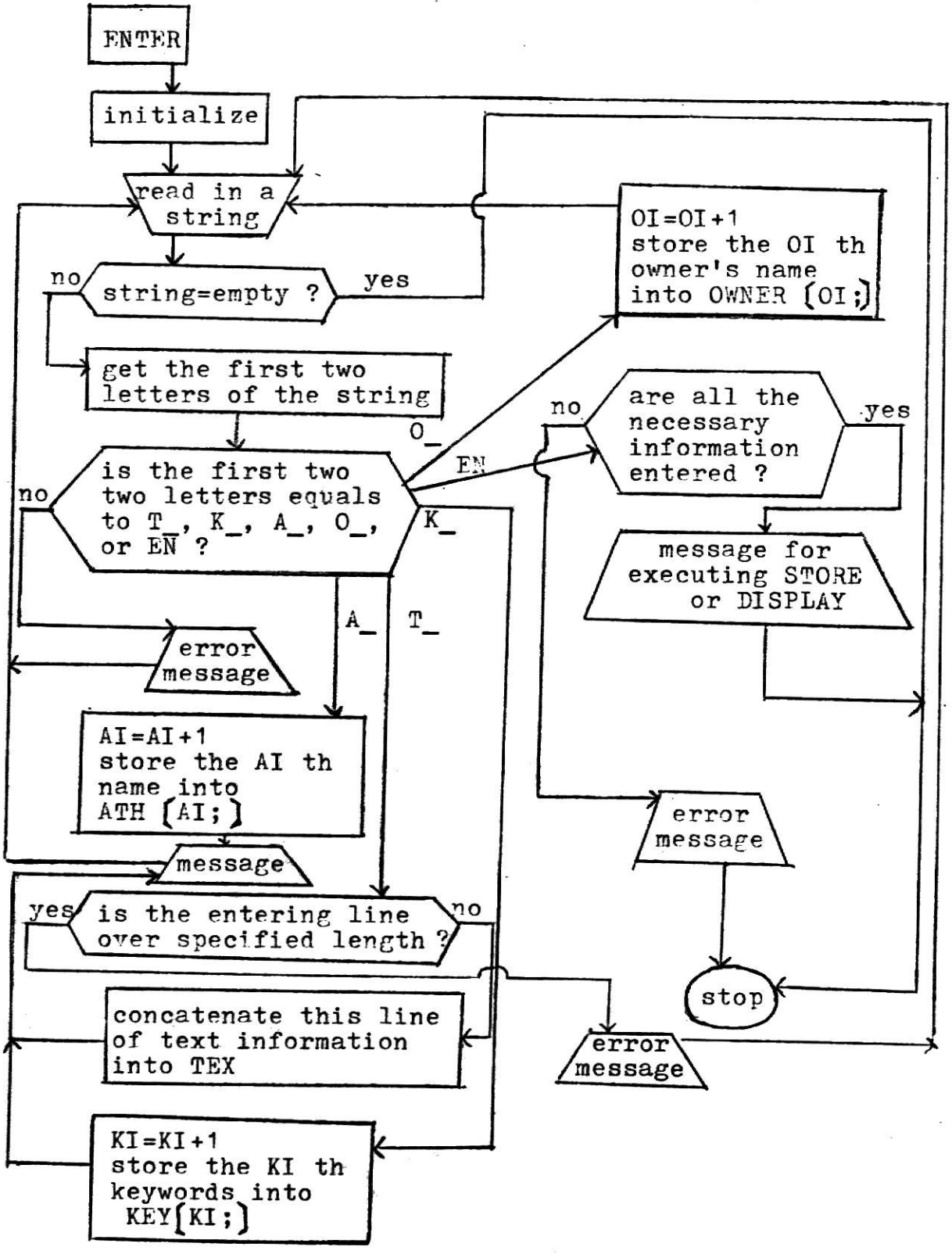
Algorithm for Deleting Keywords

Figure-A-II-15



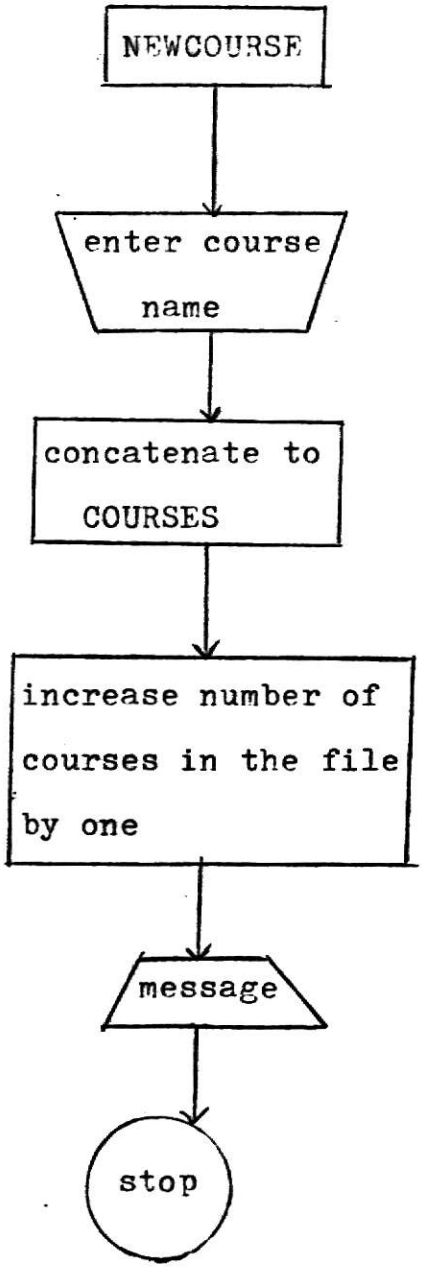
Algorithm for Entering a Document

Figure-A-II-16



Algorithm for entering a NEWCOURSE name

Figure-A-II-17



Algorithm for entering

a new owner's name

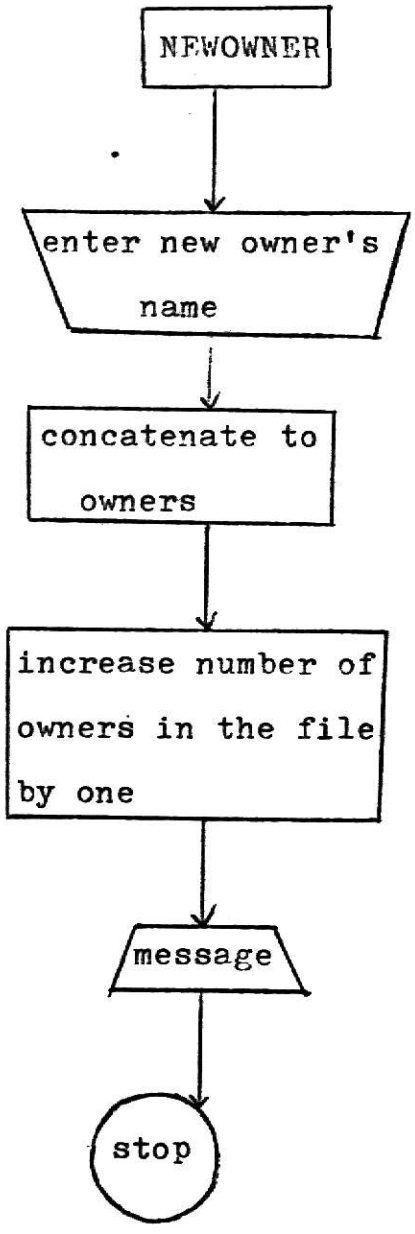
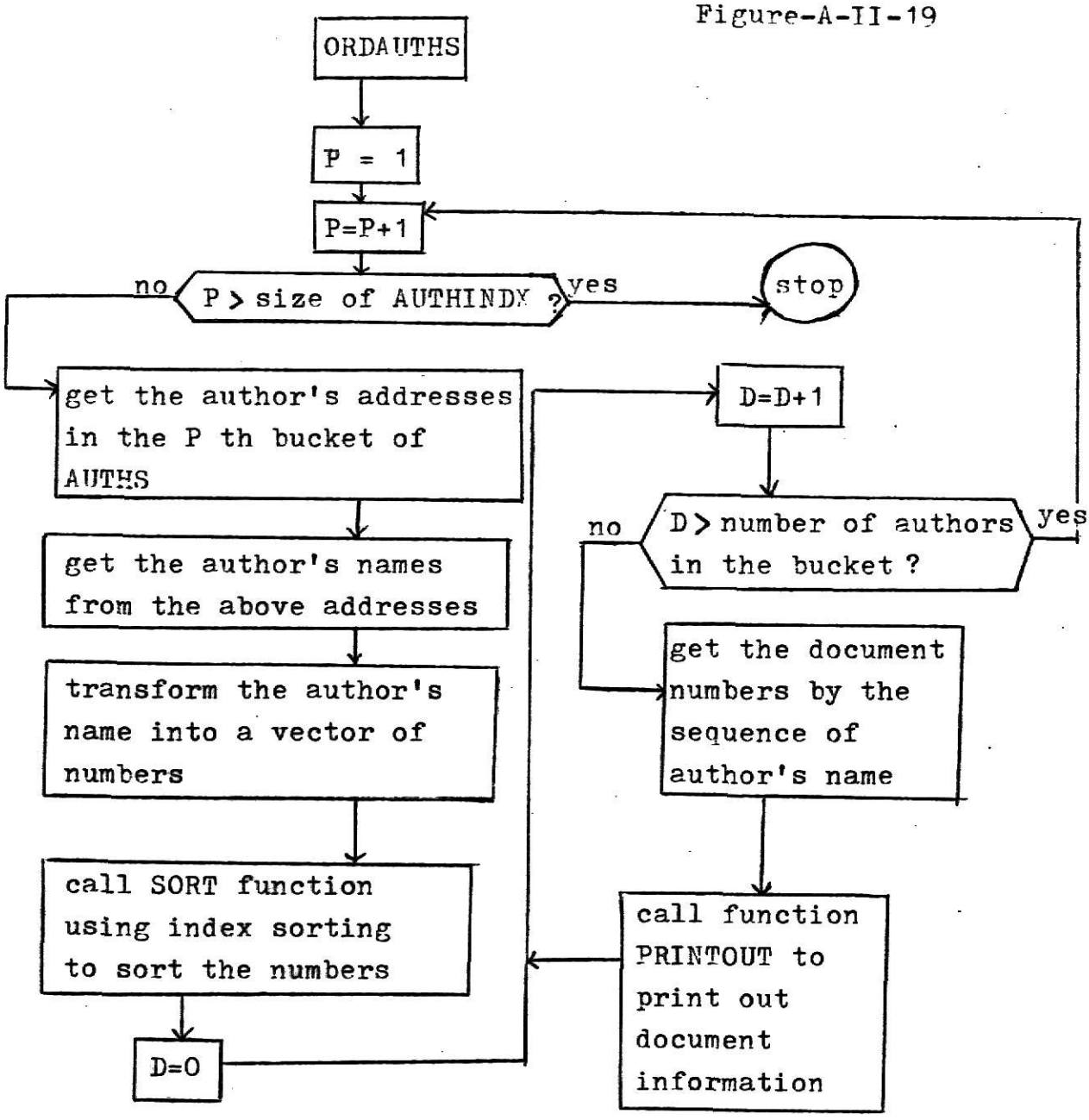


Figure-A-II-18

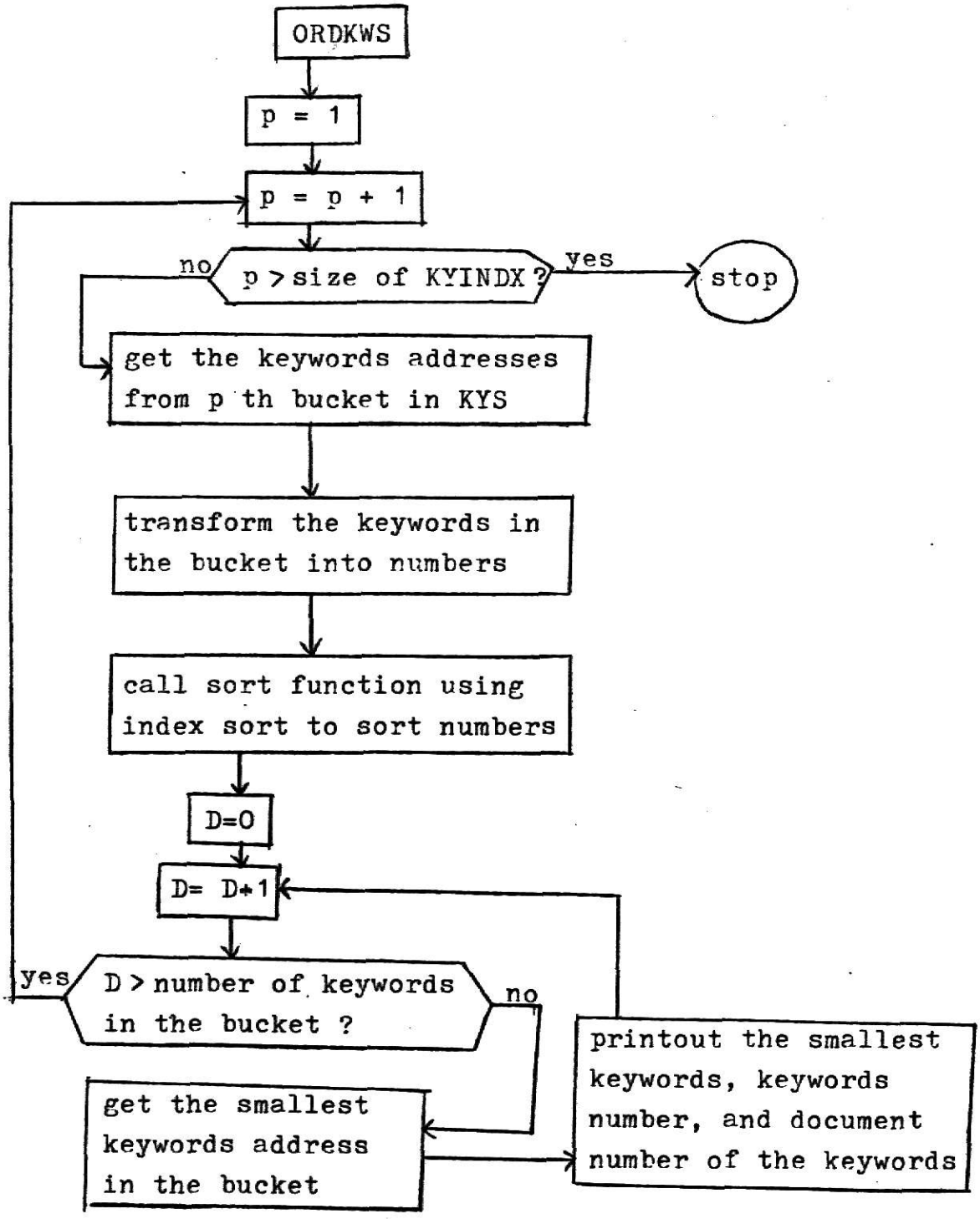
Algorithm for print out the file in alphabetical order of authors

Figure-A-II-19



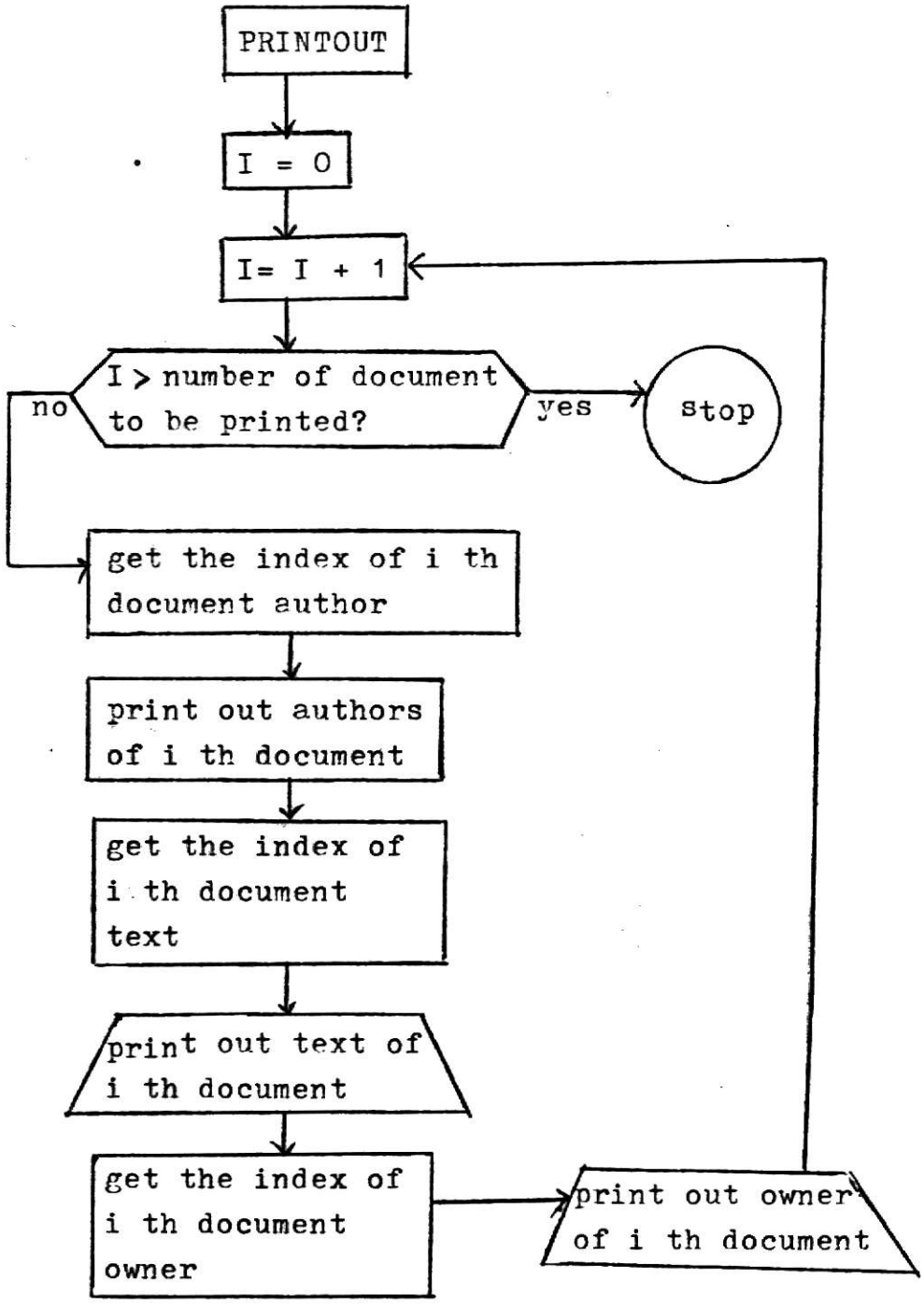
Algorithm for print out sorted keywords and document number cross references

Figure-A-II-20



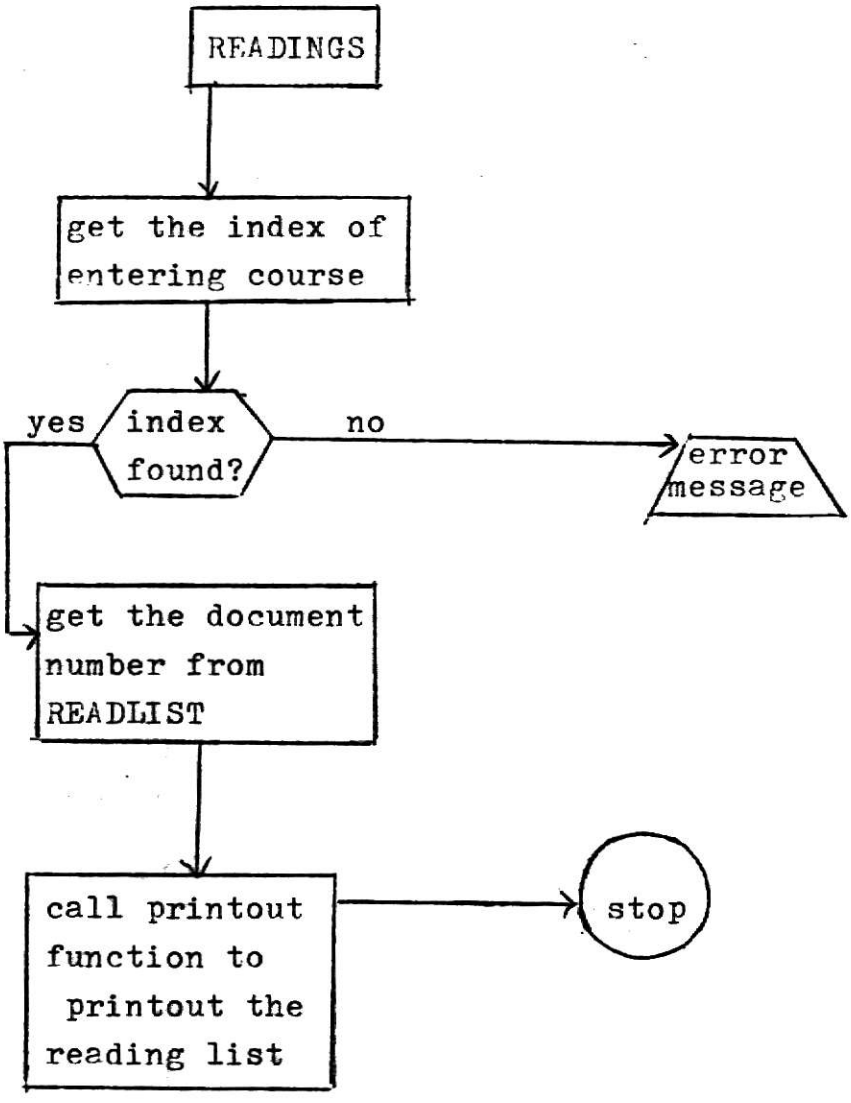
Algorithm for function
PRINTOUT

Figure-A-II-21



Algorithm for search
reading list of a course

Figure-A-II-22



Algorithm for
sorting by index

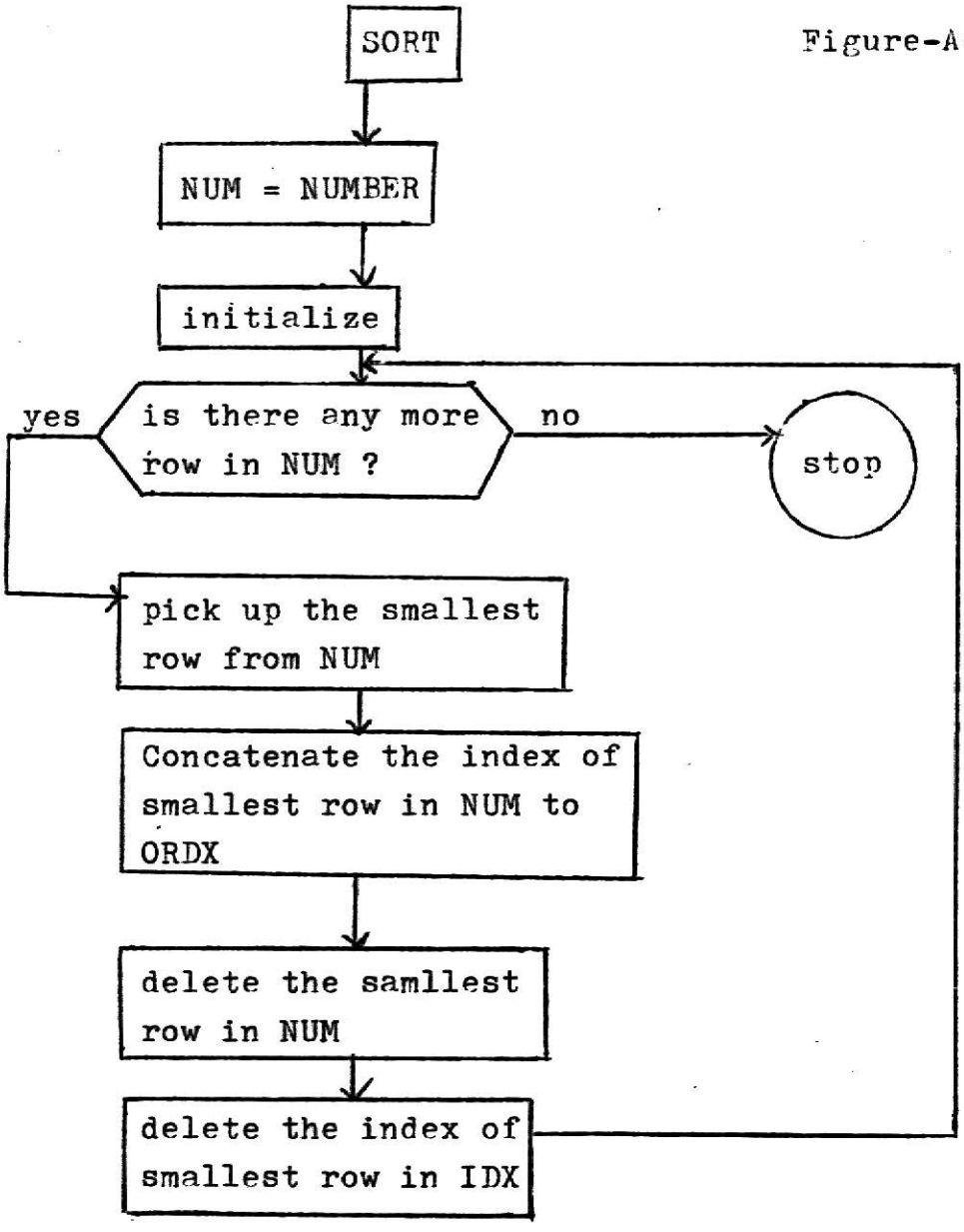
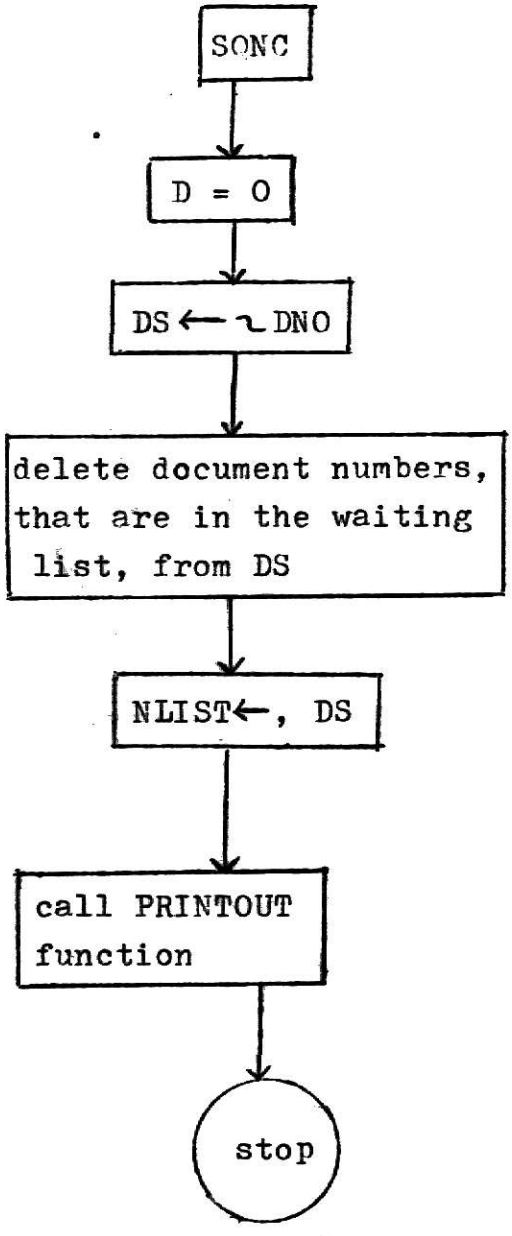


Figure-A-II-23

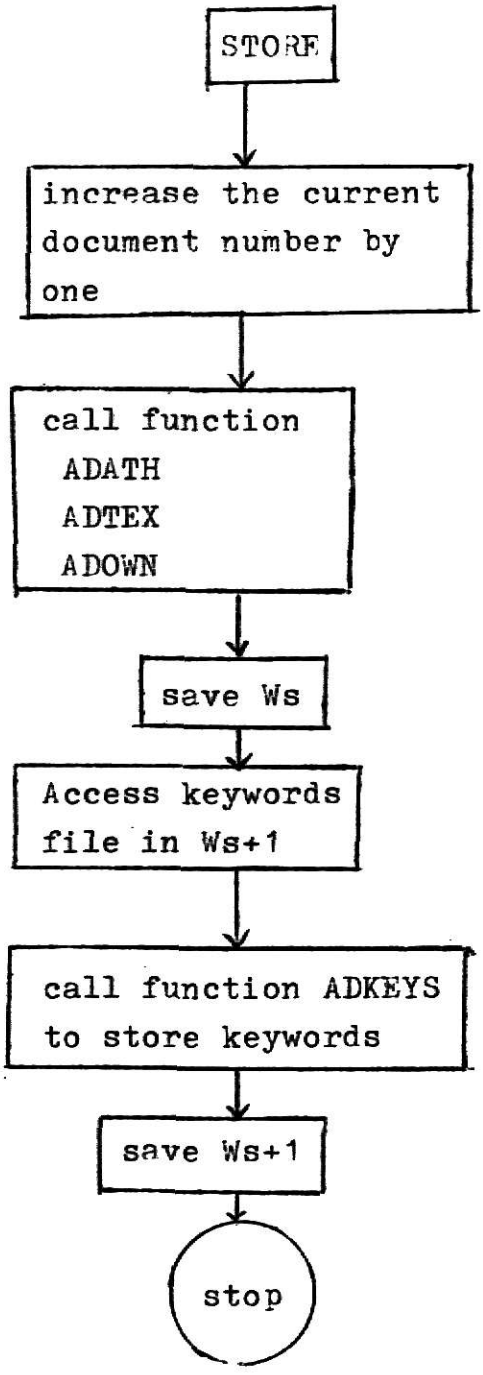
Algorithm for getting the sequence of document numbers

Figure-A-II-24



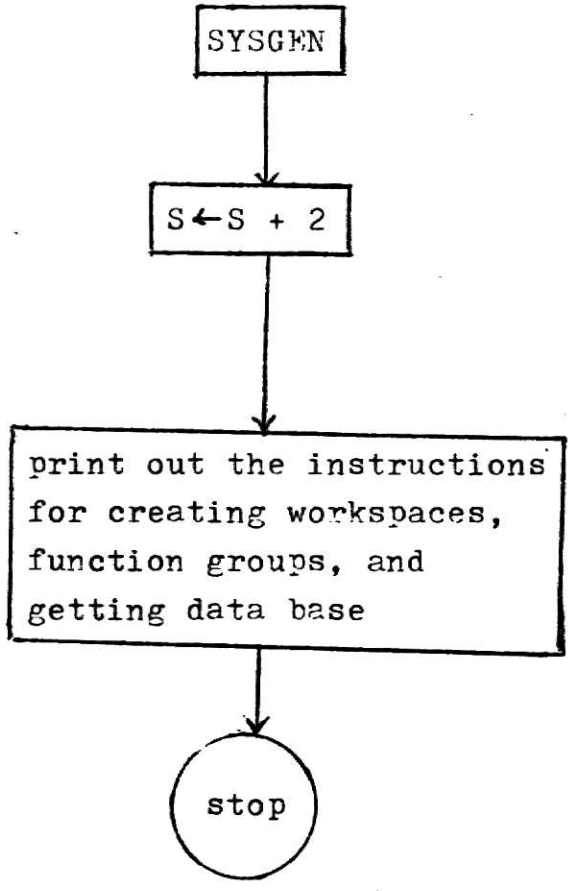
Algorithm for
function STORE

Figure-A-II-25



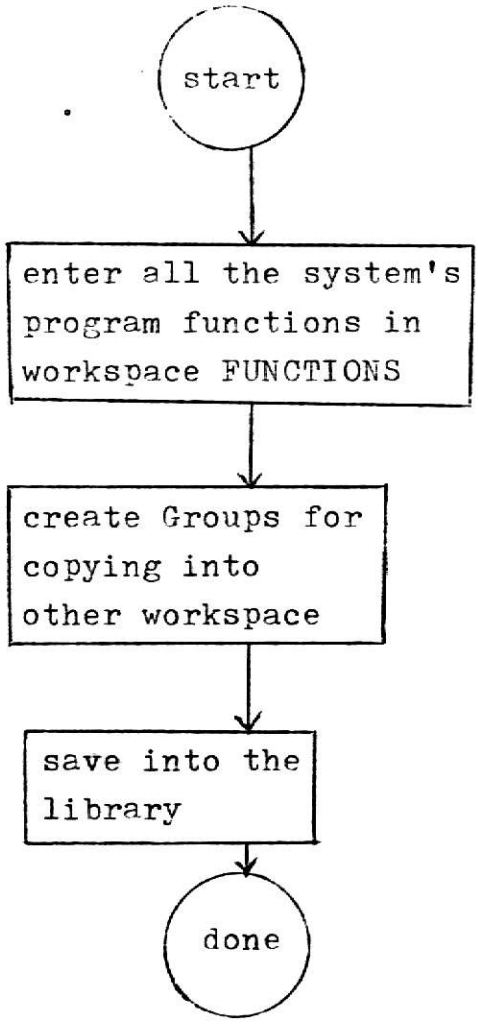
Algorithm for
function SYSGEN

Figure-A-II-26



Algorithm for creating the library named FUNCTIONS

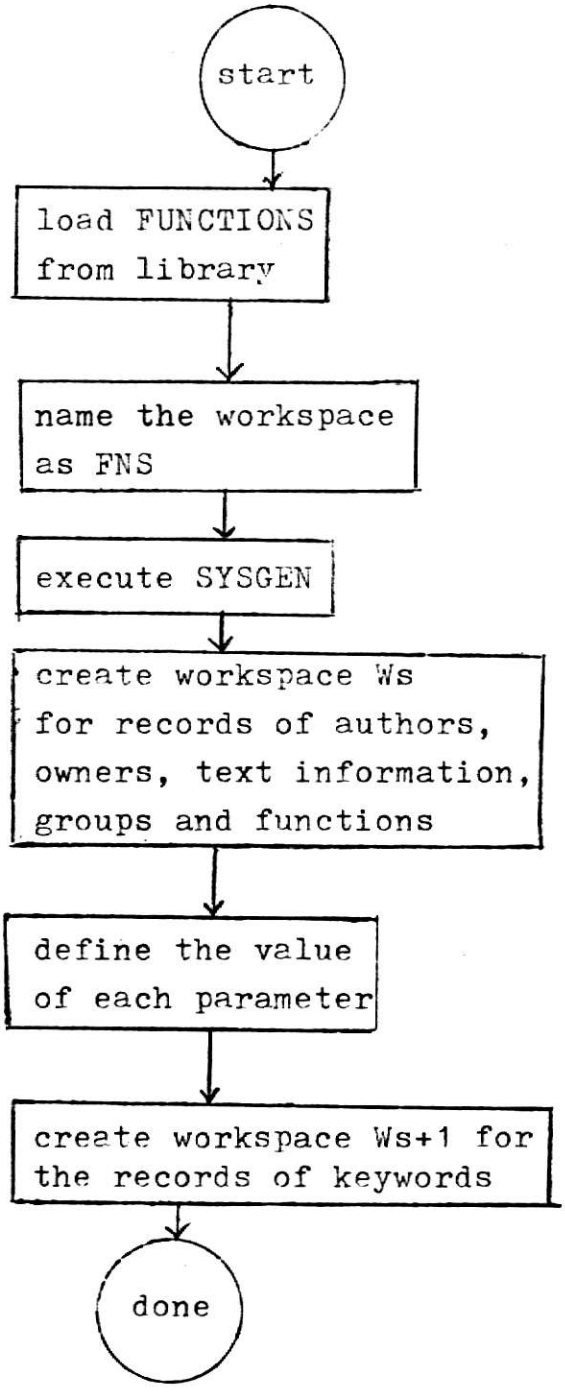
Figure-A-27



Algorithm for

System Initialization

Figure-A-II-28



D. List of Groups and Functions (see Figure-III-4 and list of programs)

```
      )GRPS
CREATION1       CREATION2       DATA1    DATA2    DUMP1
DUMP2   INIT1   INIT2   KVARS   KWS      PTA      RETRIEVE1
RETRIEVE2


      )GRP CREATION1
ENTER   ADATH   ADOWN   ADTEX   STORE   DISPLAY

      )GRP CREATION2
ADKEYS

      )GRP DATA1
AL15    AUTHBKS AUTHINDX        AUTHS   BKAUTHS BKOWNERS
COURSES DNO     LETTERS L5      L50 NC  NA      NWS
OWNERS  READLIST        STARTINDX       TEXTINDX
TEXTS   TXWAIT  S

      )GRP DATA2
KSTARTINDX      KYBKS   KYINDX  KYS     LETTERS L45     S

      )GRP DUMP1
PRINTOUT        SQNC

      )GRP DUMP2
ORDKWS  SORT

      )GRP INIT1
DEFAULT DEFINE  INITIALIZE

      )GRP INIT2
INIKFL

      )GRP KVARS
NKB     KSPACE  NB      MKB     L45     LETTERS MBX     S

      )GRP KWS
KEY     KI      BKNO

      )GRP PTA
ORDAUTHS        SORT    PRINTOUT

      )GRP RETRIEVE1
BYAUTHOR        PRINTOUT        BYBOOKNUMBER    READINGS

      )GRP RETRIEVE2
BYKEYWORD       BYKEYNUMBER     PRINTOUT
```

```
      )FNS
ADATH   ADKEYS  ADOWN   ADREADLIST      ADTEX   BYAUTHOR
BYBOOKNUMBER    BYKEYNUMBER     BYKEYWORD       COLLECT
DEFAULT DEFINE  DELETE  DISPLAY DTKEY   ENTER   INIKFL
INITIALIZE      NEWCOURSE       NEWOWNER        ORDAUTHS
ORDKWS  PRINTOUT        READINGS        SORT    SQNC
STORE   SYSGEN
```

```
      )VARS
S
```

```
[44]  [⎕]
    ∇ ADATH;I;J;P;Y;X;Z;XY;OY;SP;TEM
[1]   I←0
[2]  MRA:I←I+1
[3]   →(I>AI)/AEND
[4]   J←1
[5]   →((I>AL3-1)∧(AL3<AI))/FORTHA
[6]   →(1=P←LETTERS⍳ATH[I;1])/BLANK
[7]   Y←AUTHINDX[P]
[8]   X←STARTINDX[P]
[9]   Z←STARTINDX[P+1]
[10]  XY←(X-1)+⍳(Y-X)
[11]  →(0≠OY←(AUTHS[XY;]∧.=ATH[I;])/XY)/ETROD
[12]  AUTHS[Y;]←ATH[I;]
[13]  AUTHINDX[P]←Y+1
[14]  AUTHBKS[Y;1]←BKNO
[15]  BKAUTHS[BKNO;I]←Y
[16]  →(Z=1+Y)/REALOCAT
[17]  →MRA
[18] FORTHA:BKAUTHS[BKNO;AL3]←NA
[19]  →MRA
[20] ETROD:→(AUTHBKS[OY;J]=0)/STABNO
[21]  J←J+1
[22]  →(J>MAX)/AKOVF
[23]  →ETROD
[24] STABNO:AUTHBKS[OY;J]←BKNO
[25]  →(I>AL3)/FORTHA
[26]  BKAUTHS[BKNO;I]←OY
[27]  →MRA
[28] AKOVF:AUTHBKS←((MAX⍴1),0)⍀AUTHBKS
[29]  MAX←MAX+1
[30]  →ETROD
[31] REALOCAT:V←⍴AUTHS
[32]  SP←SPACEINDX[P]
[33]  TEM←(Y⍴1),(SP⍴0),((V[1]-Y)⍴1)
[34]  AUTHBKS←TEM⌿[1] AUTHBKS
[35]  AUTHS←TEM⌿[1] AUTHS
[36]  STARTINDX[(⍳(28-P))+P]←STARTINDX[(⍳(28-P))+P]+SP
[37]  AUTHINDX[(⍳(28-P))+P]←AUTHINDX[(⍳(28-P))+P]+SP
[38]  NA←NA+SP
[39]  BKAUTHS←BKAUTHS+(BKAUTHS>Y)×SP
[40]  →MRA
[41] AEND:'AUTHOR''S NAME HAS BEEN STORED'
[42]  →0
[43] BLANK:'THERE IS NO AUTHOR FOR DOCUMENT ';BKNO
    ∇
[44]  ∇
```

```
      ∇ADKEYS
[44]  [□]
    ∇ ADKEYS;I;J;KP;KY;KX;KZ;KXY;KOY;TEM;KPP;V
[1]   I←0
[2]  KSTAR:I←I+1
[3]   →(I>KI)/KEND
[4]   J←1
[5]   →(1=KP←LETTERS⍳KEY[I;1])/KBLANK
[6]   KY←KYINDX[KP]
[7]   KX←KSTARTINDX[KP]
[8]   KZ←KSTARTINDX[KP+1]
[9]   KXY←(KX-1)+⍳(KY-KX)
[10]  →(0≠⍴KOY←(KYS[KXY;]∧.=KEY[I;])/KXY)/KETROD
[11]  KYS[KY;]←KEY[I;]
[12]  KYINDX[KP]←KY+1
[13]  KYBKS[KY;1]←BKNO
[14]  BKYNO[BKNO;I]←KY
[15]  →(KZ=1+KY)/KALOCAT
[16]  →KSTAR
[17] KETROD:→(KYBKS[KOY;J]=0)/STKNO
[18]  J←J+1
[19]  →(J>KMAX)/KOVF
[20]  →KETROD
[21] STKNO:KYBKS[KOY;J]←BKNO
[22]  BKYNO[BKNO;I]←KOY
[23]  →KSTAR
[24] KOVF:KYBKS←((KMAX⍴1),0)\KYBKS
[25]  KMAX←KMAX+1
[26]  →KETROD
[27] KALOCAT:V←⍴KYS
[28]  TEM←(KY⍴1),(KSPACE⍴0),((V[1]-KY)⍴1)
[29]  KYBKS←TEM\[1] KYBKS
[30]  KYS←TEM\[1] KYS
[31]  KPP←KP+⍳(28-KP)
[32]  KSTARTINDX[KPP]←KSTARTINDX[KPP]+KSPACE
[33]  KYINDX[KPP]←KYINDX[KPP]+KSPACE
[34]  BKYNO←BKYNO+(BKYNO>KY)×KSPACE
[35]  →KSTAR
[36] KEND:'KEY WORDS HAVE BEEN STORED'
[37]  ⍳0
[38]  →40
[39] KBLANK:'THERE IS NO KEY WORD FOR DOCUMENT ';BKNO
[40]  'ENTER: )SAVE W';S+1
[41]  '              FOR ENTERING ANOTHER DOCUMENT TYPE:'
[42]  '     1:)LOAD W';S
[43]  '     2: ENTER'
    ∇
[44]  ∇
```

```
      ∇ADOWN
[10]  [□]
    ∇ ADOWN;I;OTH
[1]   I←0
[2]  AGN:I←I+1
[3]   →(I>OI)/FSH
[4]   →(0=ρOTH←(OWNERS∧.=OWNER[I;])/ιNWS)/WRN
[5]   BKOWNERS[BKNO;I]←OTH
[6]   →AGN
[7]  FSH:'NAME OF OWNER HAS BEEN STORED'
[8]   →0
[9]  WRN:OWNER[I;];' IS A WRONG NAME, PLEASE SEE YOUR MANAG
      ER'
    ∇
[10]  ∇
```

```
      ∇ADREADLIST
[11]  [□]
    ∇ ADREADLIST;CSNO;CS;LIST
[1]   'ENTER THE COURSE NUMBER:'
[2]   CSNO←⍞
[3]   'ENTER THE BOOK NUMBER(S)'
[4]   LIST←,□
[5]   →(0=ρCS←(COURSES∧.=CSNO)/ιNC)/NOTIN
[6]   READLIST[CS;ι⍴LIST]←LIST
[7]   'THE READING LIST FOR ';CSNO;' HAS BEEN ENTERED'
[8]   'ENTER: )SAVE W';S
[9]   →0
[10] NOTIN:'THE ENTERED COURSE NUMBER IS NOT IN W';S
    ∇
[11]  ∇
```

```
      ∇ADTEX
[4]   [□]
    ∇ ADTEX
[1]   TEXTS←TEXTS,TEX
[2]   TEXTINDX←TEXTINDX,(TEXTINDX[BKNO-1]+ρTEX)
[3]   'TEXT INFORMATION HAS BEEN STORED'
    ∇
[4]   ∇
```

```
      ∇BYAUTHOR
[18]  [□]
    ∇ BYAUTHOR;AH;AP;AY;AX;AXY;AOY;TEM
[1]   NAME:'ENTER  NAME:'
[2]   AH←⍞
[3]   →0×⍳0=⍴AH
[4]   AH←AL15↑AH
[5]   AP←LETTERS⍳X1←1↑AH
[6]   AY←AUTHINDX[AP]
[7]   AX←STARTINDX[AP]
[8]   AXY←(AX-1)+⍳(AY-AX)
[9]   →(0=⍴AOY←(AUTHS[AXY;]∧.=AH)/AXY)/NOATH
[10]  TEM←AUTHBKS[AOY;]
[11]  TEM←((⍴TEM)[2])⍴TEM
[12]  NLIST←(0≠TEM)/TEM
[13]  NLIST←,NLIST
[14]  PRINTOUT
[15]  →NAME
[16] NOATH:AH;' IS NOT IN W';S
[17]  →NAME
    ∇
[18]  ∇
```

```
      ∇BYBOOKNUMBER
[4]   [□]
    ∇ BYBOOKNUMBER
[1]   'ENTER DOCUMENT NUMBER(S)'
[2]   NLIST←,□
[3]   PRINTOUT
    ∇
[4]   ∇
```

```
      ∇BY
       ∨

      ∇BYKEYNUMBER
[16]  [⎕]
    ∇ BYKEYNUMBER;KYNOS;J;K;KYLIST;NST
[1]   NLIST←⍳0
[2]   K←0
[3]   'ENTER KEY WORDS NUMBER(S)'
[4]   KYNOS←,⎕
[5]   J←⍴KYNOS
[6]  KEYS:K←K+1
[7]   →(K>J)/OK
[8]   KYLIST←KYBKS[KYNOS[K];]
[9]   NLIST←NLIST,NST←(0≠KYLIST)/KYLIST
[10]  →KEYS
[11] OK:'ENTER: 1:)SAVE W';S+1
[12]  '        2:)LOAD W';S
[13]  '         3:)COPY FNS PRINTOUT'
[14]  '        4:)COPY W';S+1;' NLIST'
[15]  '        5: PRINTOUT'
    ∇
[16]  ∇


      ∇BYKEYWORD
[22]  [⎕]
    ∇ BYKEYWORD;STR;P;Y;X;XY;OY;TEM;X1
[1]  PHRASE:'ENTER KEY WORD(S):'
[2]   STR←⍞
[3]   →0×⍳0=⍴STR
[4]   STR←L45↑STR
[5]   P←LETTERS⍳X1←1↑STR
[6]   Y←KYINDX[P]
[7]   X←KSTARTINDX[P]
[8]   XY←(X-1)+⍳(Y-X)
[9]   →(0=⍴OY←(KYS[XY;]∧.=STR)/XY)/NOKY
[10]  TEM←KYBKS[OY;]
[11]  TEM←((⍴TEM)[2])⍴TEM
[12]  NLIST←⍵ 0≠TEM)/TEM
[13]  'KEY WORDS FOUND TYPE:'
[14]  '     1: SAVE W';S+1
[15]  '     2:)LOAD W';S
[16]  '     3:)COPY FNS PRINTOUT'
[17]  '     4:)COPY W';S+1;' NLIST'
[18]  '     5: PRINTOUT'
[19]  →0
[20] NOKY:STR;' IS NOT IN W';S+1
[21]  →PHRASE
    ∇
[22]  ∇
```

```
      ∇COLLECT
[20]  [□]
    ∇ COLLECT;LN;LS;TEM
[1]   LN←⍴TEX
[2]   →(0=⍴LS←(TXLENGTH≥LN)/⍳⍴TXLENGTH)/NOAV
[3]   BKNO←TXWAIT[LS[1]]
[4]   'DOCUMENT NO.';BKNO
[5]   TXLENGTH[LS[1]]←0
[6]   TXLENGTH←(TXLENGTH≠0)/TXLENGTH
[7]   TXWAIT←(TXWAIT≠TXWAIT[LS[1]])/TXWAIT
[8]   TEM←TEXTINDX[BKNO-1]+⍳(TEXTINDX[BKNO]-TEXTINDX[BKNO-1
      ])
[9]   TEXTS[TEM]←⊂TEM)↑TEX
[10]  'TEXT INFORMATION HAS BEEN STORED'
[11]  ADATH
[12]  ADOWN
[13]  'ENTER:'
[14]  '     1:)SAVE W';S
[15]  '     2:)LOAD W';S+1
[16]  '     3:)COPY W';S;' KWS'
[17]  '     4: ADKEYS'
[18]  →0
[19] NOAV:'NO STORAGE LOCATION AVAILABLE, ENTER:STORE'
    ∇
[20]  ∇
```

```
      ∇DEFAULT
[26]  [□]
    ∇ DEFAULT
[1]   NA←129
[2]   NB←100
[3]   NWS←5
[4]   NKB←132
[5]   NC←1
[6]   NR←10
[7]   KSPACE←4
[8]   MKB←8
[9]   AUTHINDX← 1 2 5 15 23 28 31 35 40 49 51 54 60
      65 75 78 80 86 88 94 106 110 112 114 122 124
      126 128
[10]  SPACEINDX← 0 3 10 8 5 3 4 5 9 2 3 6 5 10 3 2
      6 2 6 12 4 2 2 8 2 2 2
[11]  STARTINDX←AUTHINDX
[12]  MBX←4
[13]  L5←5
[14]  L15←10
[15]  COURSES←(NC,L5)ρ'DUMMY'
[16]  OWNERS←(NWS,L15)ρ'FISHER     HANKLY     WALLENTINEBREWE
      R     CALHOUN    '
[17]  AL15←15
[18]  OL3←3
[19]  AL3←3
[20]  BL3←3
[21]  L45←45
[22]  A4←4
[23]  'REDEFINE PARAMETER(S) AS NEEDED'
[24]  'ENTER: INITIALIZE'
[25]  L50←65
    ∇
[26]  ∇
```

```
      [□]
    ∇ DEFINE
[1]   'TYPE IN THE ESTIMATED TOTAL NUMBER OF AUTHORS TO BE STORED IN
      W';S;' : NA'
[2]   NA←□
[3]   'ENTER THE ESTIMATED TOTAL NUMBER OF DOCUMENTS TO BE STORED IN
      W';S;':NB'
[4]   NB←□
[5]   'ENTER THE TOTAL NUMBER OF OWNERS:NWS'
[6]   NWS←□
[7]   'ENTER THE MAXIMUM NUMBER OF KEYWORDS TO IN W';S;' : NKB'
[8]   NKB←□
[9]   'ENTER THE NUMBER OF COURSES TO BE USED FOR SEARCH READING LIST
       :NC'
[10]  NC←□
[11]  'ENTER THE MAXIMUM NUMBER OF READING LIST FOR A COURSE : NR'
[12]  NR←□
[13]  'ENTER THE MAXIMUM NUMBER OF AVAILABLE STORAGE LOCATIONS FOR KE
      YWORDS WITH THE SAME INITIAL LETTER:KSPACE'
[14]  KSPACE←□
[15]  'ENTER THE MAXIMUM NUMBER OF DESCRIPTORS FOR A DOCUMENT: MKB'
[16]  MKB←□
[17]  'ENTER THE 27 VALUES FOR AUTHOR INDEX STARTING LOCATION OF EACH
       LETTER:AUTHINDX'
[18]  AUTHINDX←□
[19]  'ENTER THE ESTIMATED MAXIMUM NUMBER OF DOCUMENTS ALLOWED FOR A
      KEYWORDS:MBK '
[20]  MBK←□
[21]  'ENTER THE VALUES OF AUTHINDX[I+1]-AUTHINDX[I]:SPACEINDX'
[22]  SPACEINDX←□
[23]  'ENTER THE LENGTH OF COURSE IDENTIFICATION :L5'
[24]  L5←□
[25]  'ENTER THE LENGTH OF THE OWNER''S NAME:L15'
[26]  L15←□
[27]  'ENTER THE LIST OF COURSES:COURSES'
[28]  COURSES←(NC,L5)ρ□
[29]  'ENTER THE LIST OF OWNER''S NAME :OWNERS'
[30]  OWNERS←(NWS,L15)ρ□
[31]  'ENTER THE LENGTH OF AUTHOR'' NAME:AL15'
[32]  AL15←□
[33]  'ENTER THE MAXIMUM NUMBER OF AUTHORS FOR A DOCUMENT:AL3'
[34]  AL3←□
[35]  'ENTER THE MAXIMUM NUMBER OF OWNERS FOR A DOCUMENT:OL3'
[36]  OL3←□
[37]  'ENTER THE INITIAL VALUE OF THE NUMBER OF DOCUMENTS ALLOWED FOR
       A AUTHOR:BL3'
[38]  BL3←□
[39]  'ENTER THE MAXIMUM LENGTH OF A DESCRIPTOR:L45'
[40]  L45←□
[41]  'THE MAXIMUM NUMBER OF AUTHORS FOR A DOCUMENT :A4'
[42]  A4←□
[43]  'ENTER THE LENGTH OF A LINE OF TEXT INFORMATION :L50'
[44]  L50←□
[45]  STARTINDX←AUTHINDX
[46]  INITIALIZE
    ∇
[47]  ∇
```

```
      ∇DELETE
[33]  [□]
    ∇ DELETE;BA;ANO;AU;MOV
[1]  BGND:'ENTER TO BE DELETED DOCUMENT:'
[2]   DTNO←□
[3]   →(0=DTNO)/0
[4]   TXWAIT←TXWAIT,DTNO
[5]   TXLENGTH←TXLENGTH,TEXTINDX[DTNO]-TEXTINDX[DTNO-1]
[6]   BA←(BKAUTHS[DTNO;]≠0)/BKAUTHS[DTNO;]
[7]   BKAUTHS[DTNO;]←0
[8]   BKOWNERS[DTNO;]←0
[9]   ANO←0
[10] H:ANO←ANO+1
[11]  →(ANO>⍴BA)/MESG
[12]  AU←BA[ANO]
[13]  →(AU=NA)/H
[14]  →((+/(AUTHBKS[AU;]≠0))=1)/DTAH
[15]  AUTHBKS[AU;AUTHBKS[AU;]⍳DTNO]←0
[16]  →H
[17] DTAH:P←LETTERS⍳AUTHS[AU;1]
[18]  MOV←(AU-1)+⍳(AUTHINDX[P]-AU)
[19]  AUTHS[MOV;]←AUTHS[1+MOV;]
[20]  AUTHBKS[AU;]←0
[21]  AUTHBKS[MOV;]←AUTHBKS[1+MOV;]
[22]  BKAUTHS←BKAUTHS-((BKAUTHS>AU)∧(BKAUTHS<AUTHINDX[P]))
[23]  BA←BA-((BA>AU)∧(BA<AUTHINDX[P]))
[24]  AUTHINDX[P]←AUTHINDX[P]-1
[25]  →H
[26] MESG:'AUTHOR(S) DELETED'
[27]  'ENTER:'
[28]  '     1:)SAVE W';S
[29]  '     2:)LOAD W';S+1
[30]  '     3: )COPY FNS DTKEY'
[31]  '     4: )COPY W';S;' DTNO'
[32]  '     5: )DTKEY'
    ∇
[33]  ∇
```

```
      ∇DISPLAY
[5]   [□]
    ∇ DISPLAY
[1]   ATH[⍳AI;]
[2]   TEX
[3]   OWNER[⍳OI;]
[4]   KEY[⍳KI;]
    ∇
[5]   ∇
```

```
      ∇DTKEY
[21]  [□]
    ∇ DTKEY;KA;KN;KU;KMOV
[1]   KA←(BKYNO[DTNO;]≠0)/BKYNO[DTNO;]
[2]   BKYNO[DTNO;]←0
[3]   KN←0
[4]  K:KN←KN+1
[5]   →(KN>⍴KA)/KSG
[6]   KU←KA[KN]
[7]   →((+/(KYBKS[KU;]≠0))=1)/KDT
[8]   KYBKS[KU;KYBKS[KU;]⍳DTNO]←0
[9]   →K
[10] KDT:P←LETTERS⍳KYS[KU;1]
[11]  KMOV←(KU-1)+⍳(KYINDX[P]-KU)
[12]  KYS[KMOV;]←KYS[1+KMOV;]
[13]  KYBKS[KU;]←0
[14]  KYBKS[KMOV;]←KYBKS[KMOV+1;]
[15]  BKYNO←BKYNO-((BKYNO>KU)∧(BKYNO<KYINDX[P]))
[16]  KA←KA-((KA>KU)∧(KA<KYINDX[P]))
[17]  KYINDX[P]←KYINDX[P]-1
[18]  →K
[19] KSG:'KEY WORD(S) DELETED'
[20]  'ENTER: )SAVE W';S+1
    ∇
[21]  ∇
```

```
      [□]
    ∇ ENTER;HED;STR;HEDS
[1]  MORE:HEDS←⍳0
[2]   AI←OI←KI←0
[3]   TEX←⍳0
[4]  READIN:'ENTER ONE LINE PLEASE'
[5]   STR←⍞
[6]   →0×⍳0=⍴STR
[7]   HED←2↑STR
[8]   HEDS←HEDS,1↑HED
[9]   →(TAB∧.=HED)/A1,A2,A3,AO,FINISHENTER
[10]  'ERROR:INPUT PROCEDURE WAS IMPROPER, CORRECT THE ABOVE LINE
      BY RETYPING THE RIGHT STRING'
[11]  →READIN
[12] A1:AI←AI+1
[13]  ATH[AI;]←AL15↑(2↓STR)
[14]  →READIN
[15] A2:→(L50<(⍴STR)-2)/OVER50
[16]  TEX←TEX,L50↑(2↓STR)
[17]  →READIN
[18] A3:KI←KI+1
[19]  KEY[KI;]←L45↑(2↓STR)
[20]  →READIN
[21] AO:OI←OI+1
[22]  →(OI>OL3)/TOOMANY
[23]  OWNER[OI;]←L15↑(2↓STR)
[24]  →READIN
[25] TOOMANY:'TOO MANY OWNER''S NAME WERE ENTERED, THE SYSTEM ONLY
       STORE THE FIRST ';OL3;' OWNERS'
[26]  OI←OI-1
[27]  →READIN
[28] FINISHENTER:→((+/('ATKOE'∈HEDS))<5)/ERR
[29]  'THE DATA WERE SUCCESSFULLY ENTERED,IF YOU WANT TO CHECK TYP
      ING ERROR TYPE DISPLAY,OTHERWEISE TYPE STORE'
[30]  →0
[31] ERR:'IMPROPER ENTERING PROCEDURE, CONSULT YOUR MANAGER FOR PR
      OPER ENTERING PROCEDURE'
[32]  →0
[33] OVER50:'TEXT INFORMATION OVER ';L50;' CHARACTERS,TYPE AGAIN'
[34]  STR←⍞
[35]  →A2
    ∇
[36]  ∇
```

```
      ∇INIKFL
[8]   [□]
    ∇ INIKFL
[1]   KYS←(NKB,L45)ρ' '
[2]   KSTARTINDX←1,((((ι27)×KSPACE)+1)-KSPACE)+1
[3]   KYINDX←1,KSTARTINDX[(ι27)+1]+1
[4]   KYBKS←(NKB,MBK)ρ0
[5]   KMAX←MBK
[6]   BKNO←1
[7]   BKYNO←(NKB,MKB)ρ0
    ∇
[8]   ∇
```

```
      ∇INITIALIZE
[31]  [□]
    ∇ INITIALIZE
[1]   AUTHS←(NA,AL15)ρ' '
[2]   AUTHS[NA;]←AL15↑'ETC'
[3]   AUTHBKS←(NA,BL3)ρ0
[4]   LETTERS←' ABCDEFGHIJKLMNOPQRSTUVWXYZ'
[5]   TEXTS←ι0
[6]   TEXTS←,TEXTS
[7]   TEXTINDX←0
[8]   TEXTINDX←,TEXTINDX
[9]   BKAUTHS←(NB,AL3)ρ0
[10]  BKOWNERS←(NB,OL3)ρ0
[11]  BKNO←1
[12]  READLIST←(NC,NR)ρ0
[13]  ATH←(A4,AL15)ρ' '
[14]  MAX←BL3
[15]  KEY←(MKB,L45)ρ' '
[16]  OWNER←(OL3,L15)ρ' '
[17]  TAB← 5 2 ρ'A_T_K_O_EN'
[18]  AUTHINDX←1,AUTHINDX[(ι27)+1]+1
[19]  TXWAIT←,TXLENGTH←,ι0
[20]  DNO←1
[21]  'TYPE:'
[22]  '     1:)ERASE INIT1'
[23]  '     2:)SAVE W';S
[24]  '     3: )CLEAR'
[25]  '     4:)WSID W';S+1
[26]  '     5:)COPY FNS INIT2'
[27]  '     6:)COPY W';S;' KVARS'
[28]  '     7: INIKFL'
[29]  '     8:)ERASE INIKFL'
[30]  '     9: )SAVE W';S+1
    ∇
[31]  ∇
```

```
      ∇NEWCOURSE
[8]   □
    ∇ NEWCOURSE;NEWCS
[1]   'ENTER COURSE NAME:'
[2]   NEWCS←L5↑⍞
[3]   COURSES←COURSES,[1] NEWCS
[4]   NC←NC+1
[5]   READLIST←READLIST,[1] NP⍴0
[6]   'NEW COURSE WAS ENTERED'
[7]   'ENTER: )SAVE W';S
    ∇
[8]   ∇
```

```
      ∇NEWOWNER
[7]   □
    ∇ NEWOWNER;NWO
[1]   'ENTER NEW OWNER''S NAME'
[2]   NWO←L15↑⍞
[3]   OWNERS←OWNERS,[1] NWO
[4]   NWS←NWS+1
[5]   'NEW OWNER HAS BEEN ENTERED'
[6]   'ENTER: )SAVE W';S
    ∇
[7]   ∇
```

```
      ∇ORDAUTHS
[29]  [□]
    ∇ ORDAUTHS;CHRS;Y;X;XY;IX;D;SZ;L;NUMBER;P;TEM;NLIST
[1]   10ρ' ';'LIST OF AUTHORS IN ALPHABETICAL ORDER:'
[2]   ⍳0
[3]   P←1
[4]   CHRS←' .,-ABCDEFGHIJKLMNOPQRSTUVWXYZ'
[5]  NEXT:P←P+1
[6]   →(P>ρAUTHINDX)/0
[7]   Y←AUTHINDX[P]
[8]   X←STARTINDX[P]
[9]   XY←(X-1)+⍳(Y-X)
[10]  IX←(AUTHS[XY;]∨.≠AL15↑' ')/XY
[11]  D←0
[12]  IX←(IX≠NA)/IX
[13]  SZ←ρIX
[14]  L←AL15
[15]  NUMBER←(SZ,AL15)ρ0
[16] INC:D←D+1
[17]  →(D>SZ)/SRT
[18]  NUMBER[D;]←CHRS⍳AUTHS[IX[D];⍳AL15]
[19]  →INC
[20] SRT:SORT
[21]  D←0
[22] MRE:D←D+1
[23]  →(D>SZ)/NEXT
[24]  TEM←AUTHBKS[IX[ORDX[D]];]
[25]  NLIST←,(0≠TEM)/TEM
[26]  PRINTOUT
[27]  →MRE

    ∇
[29]
```

```
      ∇ORDKWS
[27]  [□]
    ∇ ORDKWS;CHRS;P;X;Y;XY;IX;SZ;L;NUMBER;D;TH;TEM
[1]   10ρ' ';'LIST OF KEY WORDS AND DOCUMENT NUMBER(S) CROSS REF
      ERENCES:'
[2]   ⍳0
[3]   CHRS←' ,.-ABCDEFGHIJKLMNOPQRSTUVWXYZ'
[4]   P←1
[5]  NEXT:P←P+1
[6]   →(P>ρKYINDX)/0
[7]   Y←KYINDX[P]
[8]   X←KSTARTINDX[P]
[9]   XY←(X-1)+⍳(Y-X)
[10]  IX←(KYS[XY;]∨.≠L45↑' ')/XY
[11]  SZ←ρIX
[12]  L←L45-1
[13]  NUMBER←(SZ,L)ρ0
[14]  D←0
[15] ICR:D←D+1
[16]  →(D>SZ)/SRT
[17]  NUMBER[D;]←CHRS⍳KYS[IX[D];1+⍳L]
[18]  →ICR
[19] SRT:SORT
[20]  D←0
[21] MRE:D←D+1
[22]  →(D>SZ)/NEXT
[23]  TH←IX[ORDX[D]]
[24]  TEM←KYBKS[TH;]
[25]  'NO. ';TH;': ';KYS[TH;];15ρ'-';(TEM≠0)/TEM
[26]  →MRE
    ∇
[27]  ∇
```

```
      ∇PRINTOUT
[16]  [□]
    ∇ PRINTOUT;I;N;BN;ATHN;PI;PJ;LGTH;ONR
[1]   I←0
[2]   N←⍴NLIST
[3]  PMOER:→(N<I←I+1)/0
[4]   BN←NLIST[I]
[5]   ⍳0
[6]   'DOCUMENT NO. ';BN
[7]   ATHN←(0≠BKAUTHS[BN;])/BKAUTHS[BN;]
[8]   AUTHS[ATHN;]
[9]   PI←TEXTINDX[BN]
[10]  PJ←TEXTINDX[BN-1]
[11]  LGTH←⍴TEM←TEXTS[PJ+⍳(PI-PJ)]
[12]  ((LGTH÷L50),L50)⍴TEM
[13]  ONR←(0≠BKOWNERS[BN;])/BKOWNERS[BN;]
[14]  'OWNER:';OWNERS[ONR;]
[15]  →PMOER
    ∇
[16]  ∇
```

```
      ∇READINGS
[9]   [□]
    ∇ READINGS;CS;J;TEM
[1]   'ENTER COURSE NUMBER:'
[2]   CS←L5↑⍞
[3]   →(0=⍴J←(COURSES∧.=CS)/⍳NC)/NOCS
[4]   TEM←READLIST[J;]
[5]   NLIST←,(0≠TEM←((⍴TEM)[2])⍴TEM)/TEM
[6]   PRINTOUT
[7]   →0
[8]  NOCS:CS;' READING LIST IS NOT IN W';S
    ∇
[9]   ∇
```

```
      ∇SORT
[25]  [□]
    ∇ SORT;IDX;NUM;J;M;V1;V2;KK;X;Y;B;OE;IDX
[1]   NUM←NUMBER
[2]   ORDX←0⍴IDX←⍳(⍴NUMBER)[1]
[3]  FH:→(0≥M←(⍴NUM)[1])/0
[4]   J←1
[5]   B←NUM[1;]
[6]  AJ:J←J+1
[7]   →(J>M)/ONE
[8]   V1←B<NUM[J;]
[9]   V2←NUM[J;]<B
[10]  KK←X←Y←0
[11] CC:KK←KK+1
[12]  →(KK>L)/OR
[13]  X←X+V1[KK]×2*L-KK
[14]  Y←Y+V2[KK]×2*L-KK
[15]  →CC
[16] OR:→(X<Y)/BONE
[17]  →AJ
[18] BONE:B←NUM[J;]
[19]  →AJ
[20] ONE:OE←NUM∧.=B
[21]  ORDX←ORDX,OE/IDX
[22]  NUM←(~OE)/[1] NUM
[23]  IDX←(~OE)/IDX
[24]  →FH
    ∇
[25]  ∇
```

```
      ∇SQNC
[11]  [□]
    ∇ SQNC;DS;DN;D
[1]   D←0
[2]   DS←⍳DNO
[3]   DS←(DS≠1)/DS
[4]   DN←⍴TXWAIT
[5]  AGN:D←D+1
[6]   →(D>DN)/NMR
[7]   DS←(DS≠TXWAIT[D])/DS
[8]   →AGN
[9]  NMR:NLIST←,DS
[10]  PRINTOUT
    ∇
[11]  ∇
```

```
      ∇STORE
[11]  [□]
    ∇ STORE
[1]   DNO←DNO+1
[2]   BKNO←DNO
[3]   'DOCUMENT NO. ';BKNO
[4]   ADATH
[5]   ADTEX
[6]   ADOWN
[7]   'ENTER:)SAVE W';S
[8]   '       )LOAD W';S+1
[9]   '       )COPY W';S;' KWS'
[10]  '       ADKEYS'
    ∇
[11]  ∇
```

```
      ∇
       [□]
    ∇ SYSGEN
[1]   S←S+2
[2]   'THE SYSGEN PROCEDURE CREATES THREE WORKSPACES :'
[3]   ⍳0
[4]   '       1)----FNS----CONTAINS BACKUP COPIES OF ALL GROUPS A
      ND FUNCTIONS'
[5]   '       2)----W';S;'----CONTAINS AUTHOR, TEXT, AND READING
      LIST DATA BASES'
[6]   '       3)----W';S+1;'----CONTAINS KEYWORD DATA BASE'
[7]   ⍳0
[8]   ⍳0
[9]   'ADDITIONAL DATABASE WORKSPACE PAIRS MAY ALSO BE CREATED(T
      HAT IS W';S+2;' AND ';'W';S+3;' )'
[10]  ⍳0
[11]  'SYSGEN ALSO INITIALIZES THE DATABASES IF DESIRED'
[12]  ⍳0
[13]  ⍳0
[14]  'THE FUNCTIONS GROUPS ARE:'
[15]  ⍳0
[16]  '       INIT1--------INITIALIZATION FUNCTIONS FOR W';S;' (O
      R W';S+2;' ETC)'
[17]  ⍳0
[18]  '       RETRIEVE1----RETRIEVE-BY-AUTHOR, RETRIEVE-BY-BOOKNU
      MBER, AND RETRIEVE-BY-READINGLIST, FOR W';S
[19]  ⍳0
[20]  '       DUMP1--------DISPLAY DATA BASE, FOR W';S
[21]  ⍳0
[22]  '       CREATION1----INFORMATION INPUT FUNCTIONS, FOR W';S
[23]  ⍳0
[24]  '       INIT2--------INITIALIZATION FUNCTIONS FOR, W';S+1;'
       (OR W';S+3;', ETC)'
[25]  ⍳0
[26]  '       RETRIEVE2----RETRIEVE-BY-KEYWORD, RETRIEVE-BY-KEYWO
      RDNUMBER, FOR W';S+1
[27]  ⍳0
[28]  '       DUMP2--------DISPLAY KEYWORD DATABASE, FOR W';S+1
[29]  ⍳0
[30]  '       CREATION2----INFORMATION FUNCTIONS FOR W';S+1
[31]  ⍳0
[32]  ⍳0
[33]  'THE DATABASE GROUPS ARE:'
[34]  ⍳0
[35]  '       DATA1---------MUST BE IN W';S
[36]  '       DATA2---------MUST BE IN W';S+1
[37]  '       KWS---------MUST BE IN BOTH W';S;' AND W';S+1;'(IN
      W';S+1;', IT MUST BE COPIED FROM W';S;' )'
[38]  '       KVARS------------------------ DITTO'
```

```
[39]   ⍳0
[40]   'INSTRUCTIONS --USE ANY OF THE FOLLOWING AS NEEDED:'
[41]   ⍳0
[42]   '       )WSID W';S;'----------------- TO CREATE AN INITIAL
       W';S
[43]   '       )COPY FNS GROUPNAME ----------TO GET FUNCTIONS OR I
       NITIALIZED DATA'
[44]   '       )ERASE GROUPNAME----------- --TO DELETE FUNCTIONS'
[45]   '       )COPY NUMBER WS DATA1---------TO USE OTHER DATA (S=
       1,3,5,---,2N-1)'
[46]   '       )COPY NUMBER WS DATA2---------TO USE OTHER DATA (S=
       2,4,6,----,2N)'
[47]   '       )SAVE WS -------------------- TO SAVE WS(S=';S;' OR
        ';S+1;')'
[48]   '       )LOAD WS -------------------- TO LOAD WS(S=';S;' OR
        ';S+1;')'
[49]   ⍳0
[50]   ⍳0
[51]   'TO INITIALIZE THE DATA BASES ENTER : DEFAULT(OR DEFINE)'
    ∇
[52]   ∇
```

Appendix III

APL Instruction for Using ILCRS

instruction	definition	example
X ← Y	assign Y to X	ATH ← 'SAWYER, T.' assign the name -- 'SAWYER, T.' to variable -- ATH
X ↑ Y	take the first X elements of Y	NAME ← 15 ↑ 'NIXSON, R.' assign 'Nixson, R.ϕϕϕϕϕ' to variable NAME
A [I;]	I th row of two dimession array A	if A is 2 by 3 array with content $\begin{matrix}ABC\\DEF\end{matrix}$, then the content of A [2;] is DEF
A [I;J]	element of I th row and J the column	
DIAL APL	sign-on APL terminal	
)number	enter account number	
V [I]	I th element of vector V	
V [2 3]	2 th & 3 th element of vector V	
A [1;2 3]	second and third element of first row	

)WSID	to name a workspace	)WSID W3
)LOAD wsname	to load a workspace into active area and replace active workspace with a copy of a stored workspace	)LOAD W1
)SAVE wsname	to save an active workspace into library	)SAVE W2
)COPY wsname object	to copy a program, group, or variable into the active workspace	)COPY FNS CREATION1
)COPY number wsname	to copy another user's workspace	)COPY 116087 FNS
)COPY number wsname object	to copy a program, group, or variable from another user's library	)COPY 116087 W1 DATA1
)LOAD number wsname	to get a copy of another user's workspace	
)ERASE objects	to erase one or more objects from the active workspace	)ERASE DATA1)ERASE DELETE CREATION1
)CLEAR	to start over with a new, blank workspace	)CLEAR
)OFF	to sign off the APL terminal	)OFF

)CONTINUE	to sign off the APL terminal with the active workspace saved	
function-name	to execute a function	ENTER
carriabe return	to tell the computer that you have completed the typing of an instruction	
press the ATTN key ('V' will be printed)	to correct the error characters which start at the typed ATTN key symbol	)LOAK WS V D W1 KANSSA STATE V AS STATE

Appendix IV

DEMONSTRATION EXAMPLES

A. Physical File Maintenance Examples

1. List of documents, for W3 (page 115)
2. List of authors in alphabetical order, for W3 (page 117)
3. List of descriptors in alphabetical order, for W4 (page 119)

B. Operation Examples

1. Retrieval by name (page 120)
2. Retrieval by document numbers (page 121)
3. Addition of a new course name (page 122)
4. Addition of a course reading list (page 122)
5. Course reading list retrieval (page 122)
6. Retrieval by descriptor (page 123)
7. Retrieval by descriptor numbers (page 124)
8. Document deletion (page 124)
9. Document entry (page 128)
10. Garbage collection (page 131)
11. Other search methods (page 134)

SQNC

DOCUMENT NO. 2
MARTIN, J.
SYSTEMS ANALYSIS FOR DATA TRANSMISSION. PRENTICE-HALL,
ENGLEWOOD CLIFFS, N. J., 1972, 909 PP.
OWNER:
BREWER

DOCUMENT NO. 3
DIEMER, A.
HENRICHS, N.
COMPUTERIZED STORAGE AND DIALOG-RETRIEVAL OF PHILOSOPHICAL
INFORMATION. COMPUTER AND THE HUMANITIES, VOL 5,NO.5,4/74,P.135
OWNER:
FISHER
HANKLY

DOCUMENT NO. 4
VEANER, A.
BALLOTS-BIBLIOGRAPHIC AUTOMATION OF A LARGE LIBRARY USING A
TIME-SHARING SYSTEM. COMPUTERS AND HUMANITIES,VOL 5,NO.5,5/71
OWNER:
CALHOUN

DOCUMENT NO. 5
SHAMIR, E.
SOME INHERENT AMBIGUITY OF CERTAIN CONTEXT-FREE LANGUAGES,
DEP. OF MATH.,HEBREW U, JERUSALEM,ISRAEL,INFOR AND CONTROL,VOL.
18,NO. 4, MAY 1971, PP.355-363
OWNER:
HANKLY

DOCUMENT NO. 6
FLECK, A.
ON THE COMBINATORIAL COMPLEXITY OF CONTEXT-FREE GRAMMARS.
IN PROC. OF THE IFIP CONGRES 71, '72,59-60
OWNER:
WALLENTINE

DOCUMENT NO. 7
FLECK, A.
TOWARD A THEORY OF DATA STRUCTURES, J.COMPUTER AND SYSTEMS
SCIENCE, OCT. '71,475-488
OWNER:
BREWER

DOCUMENT NO. 8
ELLIS, C.
THE HALTING PROBLEM FOR PROBABILISTIC CONTEXT-FREE GENERATORS.
J. ACM. 19,(*JULY* 1972)396-399
OWNER:
FISHER

DOCUMENT NO. 9
AHO, A.
DENINING, P.
WEAK AND MIXED STRATEGY PRECEDENCE PARSING.,J. ACM 19,2(4/72)
OWNER:
FISHER

DOCUMENT NO. 10
AHO, A.
ULLMAN, J.
THEORY OF PARSING TRANSLATION, AND COMPILING. VOL 1:*PARSING*
OWNER:
HANKLY
FISHER

DOCUMENT NO. 11
HOPCROFT, J.
ULLMAN, J.
FORMAL LANGUAGES AND THEIR RELATION TO AUTOMATA.,ADISON-WESLEY.
OWNER:
HANKLY

ORDAUTHS

DOCUMENT NO. 9
AHO, A.
DENINING, P.
WEAK AND MIXED STRATEGY PRECEDENCE PARSING.,J. ACM 19,2(4/72)
OWNER:
FISHER

DOCUMENT NO. 10
AHO, A.
ULLMAN, J.
THEORY OF PARSING TRANSLATION, AND COMPILING. VOL 1:*PARSING*
OWNER:
HANKLY
FISHER

DOCUMENT NO. 9
AHO, A.
DENINING, P.
WEAK AND MIXED STRATEGY PRECEDENCE PARSING.,J. ACM 19,2(4/72)
OWNER:
FISHER

DOCUMENT NO. 3
DIEMER, A.
HENRICHS, N.
COMPUTERIZED STORAGE AND DIALOG-RETRIEVAL OF PHILOSOPHICAL INFORMATION. COMPUTER AND THE HUMANITIES, VOL 5,*NO.*5,4/74,*P.*135
OWNER:
FISHER
HANKLY

DOCUMENT NO. 8
ELLIS, C.
THE HALTING PROBLEM FOR PROBABILISTIC CONTEXT-FREE GENERATORS.
J. ACM. 19,(*JULY* 1972)396-399
OWNER:
FISHER

DOCUMENT NO. 6
FLECK, A.
ON THE COMBINATORIAL COMPLEXITY OF CONTEXT-FREE GRAMMARS.
IN PROC. OF THE IFIP CONGRES 71, '72,59-60
OWNER:
WALLENTINE

DOCUMENT NO. 7
FLECK, A.
TOWARD A THEORY OF DATA STRUCTURES, J.COMPUTER AND SYSTEMS SCIENCE, OCT. '71,475-488
OWNER:
BREWER

DOCUMENT NO. 3
DIEMER, A.
HENRICHS, N.
COMPUTERIZED STORAGE AND DIALOG-RETRIEVAL OF PHILOSOPHICAL INFORMATION. COMPUTER AND THE HUMANITIES, VOL 5,NO.5,4/74,P.135
OWNER:
FISHER
HANKLY

DOCUMENT NO. 11
HOPCROFT, J.
ULLMAN, J.
FORMAL LANGUAGES AND THEIR RELATION TO AUTOMATA.,ADISON-WESLEY.
OWNER:
HANKLY

DOCUMENT NO. 2
MARTIN, J.
SYSTEMS ANALYSIS FOR DATA TRANSMISSION. PRENTICE-HALL, ENGLEWOOD CLIFFS, N. J., 1972, 909 PP.
OWNER:
BREWER

DOCUMENT NO. 5
SHAMIR, E.
SOME INHERENT AMBIGUITY OF CERTAIN CONTEXT-FREE LANGUAGES, DEP. OF MATH.,HEBREW U, JERUSALEM,ISRAEL,INFOR AND CONTROL,VOL. 18,NO. 4, MAY 1971, PP.355-363
OWNER:
HANKLY

DOCUMENT NO. 10
AHO, A.
ULLMAN, J.
THEORY OF PARSING TRANSLATION, AND COMPILING. VOL 1:PARSING
OWNER:
HANKLY
FISHER

DOCUMENT NO. 11
HOPCROFT, J.
ULLMAN, J.
FORMAL LANGUAGES AND THEIR RELATION TO AUTOMATA.,ADISON-WESLEY.
OWNER:
HANKLY

DOCUMENT NO. 4
VEANER, A.
BALLOTS-BIBLIOGRAPHIC AUTOMATION OF A LARGE LIBRARY USING A TIME-SHARING SYSTEM. COMPUTERS AND HUMANITIES,VOL 5,NO.5,5/71
OWNER:
CALHOUN

ORDKWS

LIST OF KEY WORDS AND DOCUMENT NUMBER(S) CROSS REFERENCES:

NO. 3: *AUTOMATA* ----------11
NO. 12: *CONTEXT-FREE GRAMMARS* ----------6
NO. 11: *CONTEXT-FREE LANGUAGES* ----------5 7
NO. 16: *DATA STRUCTURES* ----------7
NO. 15: *DATA TRANSMISSION* ----------2
NO. 23: *FORMAL LANGUAGES* ----------5 6 10 11 9
NO. 36: *INFORMATION RETRIEVAL* ----------3 4
NO. 35: *INFORMATION STORAGE* ----------3
NO. 47: *LIBRARIES AND INFORMATION CENTER* ----------4
NO. 51: *MIXED STRATEGY PRECEDENCE PARSING* ----------9
NO. 55: *NET WORK GRAMMARS* ----------7
NO. 64: *PARSING* ----------10
NO. 65: *PROBABILISTIC CONTEXT-FREE GENERATORS* ----------8
NO. 63: *PROBABILISTIC GRAMMARS* ----------8
NO. 75: *RETRIEVAL AND DISSEMINATION* ----------3
NO. 79: *SYSTEM ANALYSIS* ----------2
NO. 95: *WEAK PRECEDENCE PARSING* ----------9

```
      )LOAD W3
SAVED  21.50.10 04/11/74
      )COPY FNS RETRIEVE1
SAVED  19.45.32 04/10/74


      BYAUTHOR
ENTER  NAME:
ULLMN
ULLMN            IS NOT IN W3
ENTER  NAME:
ULLMAN, J.

DOCUMENT NO. 10
AHO, A.
ULLMAN, J.
THEORY OF PARSING TRANSLATION, AND COMPILING. VOL 1:PARSING
OWNER:
HANKLY
FISHER

DOCUMENT NO. 11
HOPCROFT, J.
ULLMAN, J.
FORMAL LANGUAGES AND THEIR RELATION TO AUTOMATA.,ADISON-WESLEY.
OWNER:
HANKLY
ENTER  NAME:
FLECK, A.

DOCUMENT NO. 6
FLECK, A.
ON THE COMBINATORIAL COMPLEXITY OF CONTEXT-FREE GRAMMARS.
IN PROC. OF THE IFIP CONGRES 71, '72,59-60
OWNER:
WALLENTINE

DOCUMENT NO. 7
FLECK, A.
TOWARD A THEORY OF DATA STRUCTURES, J.COMPUTER AND SYSTEMS
SCIENCE, OCT. '71,475-488
OWNER:
BREWER
ENTER  NAME:
```

```
      BYBOOKNUMBER
ENTER DOCUMENT NUMBER(S)
□:
      2

DOCUMENT NO. 2
MARTIN, J.
SYSTEMS ANALYSIS FOR DATA TRANSMISSION. PRENTICE-HALL,
ENGLEWOOD CLIFFS, N. J., 1972, 909 PP.
OWNER:
BREWER


      BYBOOKNUMBER
ENTER DOCUMENT NUMBER(S)
□:
      10 3 7

DOCUMENT NO. 10
AHO, A.
ULLMAN, J.
THEORY OF PARSING TRANSLATION, AND COMPILING. VOL 1:PARSING
OWNER:
HANKLY
FISHER

DOCUMENT NO. 3
DIEMER, A.
HENRICHS, N.
COMPUTERIZED STORAGE AND DIALOG-RETRIEVAL OF PHILOSOPHICAL
INFORMATION. COMPUTER  AND THE HUMANITIES, VOL 5,NO.5,4/74,P.135
OWNER:
FISHER
HANKLY

DOCUMENT NO. 7
FLECK, A.
TOWARD A THEORY OF DATA STRUCTURES, J.COMPUTER AND SYSTEMS
SCIENCE, OCT. '71,475-488
OWNER:
BREWER
```

```
      COURSES
CS400
CS440
CS700
CS740
CS800
      )COPY FNS NEWCOURSE
SAVED  19.45.32 04/10/74

      NEWCOURSE
ENTER COURSE NAME:
CS798
NEW COURSE WAS ENTERED
ENTER: )SAVE W3


      ADREADLIST
ENTER THE COURSE NUMBER:
CS798
ENTER THE BOOK NUMBER(S)
□:
      11 9
THE READING LIST FOR CS798 HAS BEEN ENTERED
ENTER: )SAVE W3

      )SAVE W3
  15.27.29 05/15/74



      READINGS
ENTER COURSE NUMBER:
CS798

DOCUMENT NO. 11
HOPCROFT, J.
ULLMAN, J.
FORMAL LANGUAGES AND THEIR RELATION TO AUTOMATA.,ADISON-WESLEY.
OWNER:
HANKLY

DOCUMENT NO. 9
AHO, A.
DENINING, P.
WEAK AND MIXED STRATEGY PRECEDENCE PARSING.,J. ACM 19,2(4/72)
OWNER:
FISHER
```

```
      )COPY FN
           V
           W4 DATA2
SAVED  21.52.29 04/11/74

      BYKEYWORD
ENTER KEY WORD(S):
INFORMATION RETRIEVAL
KEY WORDS FOUND TYPE:
     1:)SAVE W4
     2:)LOAD W3
     3:)COPY FNS PRINTOUT
     4:)COPY W4 NLIST
     5: PRINTOUT

      PRINTOUT

DOCUMENT NO. 3
DIEMER, A.
HENRICHS, N.
COMPUTERIZED STORAGE AND DIALOG-RETRIEVAL OF PHILOSOPHICAL
INFORMATION. COMPUTER  AND THE HUMANITIES, VOL 5,NO.5,4/74,P.135
OWNER:
FISHER
HANKLY

DOCUMENT NO. 4
VEANER, A.
BALLOTS-BIBLIOGRAPHIC AUTOMATION OF A LARGE LIBRARY USING A
TIME-SHARING SYSTEM. COMPUTERS AND HUMANITIES,VOL 5,NO.5,5/71
OWNER:
CALHOUN


      BYKEYWORD
ENTER KEY WORD(S):
NOT SAVED KEYWORD
NOT SAVED KEYWORD                                IS NOT IN W4

ENTER KEY WORD(S):
```

```
      BYKEYNUMBER
ENTER KEY WORDS NUMBER(S)
□:
      2 3
ENTER: 1:)SAVE W4
       2:)LOAD W3
        3:)COPY FNS PRINTOUT
       4:)COPY W4 NLIST
       5: PRINTOUT


      PRINTOUT

DOCUMENT NO. 11
HOPCROFT, J.
ULLMAN, J.
FORMAL LANGUAGES AND THEIR RELATION TO AUTOMATA.,ADISON-WESLEY.
OWNER:
FANKLY




       )LOAD W3
SAVED   15.27.29 05/15/74
       )COPY FNS DELETE
SAVED   19.45.32 04/10/74
      DELETE
ENTER TO BE DELETED DOCUMENT:
□:
       3
AUTHOR(S) DELETED
ENTER:
     1:)SAVE W3
     2:)LOAD W4
     3: )COPY FNS DTKEY (AS NEEDED)
     4: )COPY W3 DTNO
     5: )DTKEY
       )SAVE W3
  15.53.59 05/15/74
       )LOAD W4
SAVED   21.52.29 04/11/74
       )COPY FNS DTKEY
SAVED   19.45.32 04/10/74
       )COPY W3 DTNO
SAVED   15.53.59 05/15/74
      DTKEY
KEY WORD(S) DELETED
ENTER: )SAVE W4
       )SAVE W4
  15.55.25 05/15/74
```

```
      )LOAD W3
SAVED  16.02.03 05/15/74
      DELETE
ENTER TO BE DELETED DOCUMENT:
□:
      7
AUTHOR(S) DELETED
ENTER:
    1:)SAVE W3
    2:)LOAD W4
    3: )COPY FNS DTKEY (AS NEEDED)
    4: )COPY W3 DTNO
    5: )DTKEY
      DELETE
ENTER TO BE DELETED DOCUMENT:
□:
      10
AUTHOR(S) DELETED
ENTER:
    1:)SAVE W3
    2:)LOAD W4
    3: )COPY FNS DTKEY (AS NEEDED)
    4: )COPY W3 DTNO
    5: )DTKEY


      )SAVE W3
  16.16.10 05/15/74

      )LOAD W4
SAVED  15.55.25 05/15/74
      DTNO←7
      DTKEY
KEY WORD(S) DELETED
ENTER: )SAVE W4
      DTNO←10
      DTKEY
KEY WORD(S) DELETED
ENTER: )SAVE W4
      )SAVE W4
  16.17.31 05/15/74


     )COPY FNS DUMP2
SAVED  19.45.32 04/10/74
```

```
      )LOAD W3
SAVED  16.16.10 05/15/74
      )COPY FNS SQN
                v
                DUMP1
SAVED  19.45.32 04/10/74
      SQNC
```

DOCUMENT NO. 2
MARTIN, J.
SYSTEMS ANALYSIS FOR DATA TRANSMISSION. PRENTICE-HALL, ENGLEWOOD CLIFFS, N. J., 1972, 909 PP.
OWNER:
BREWER

DOCUMENT NO. 4
VEANER, A.
BALLOTS-BIBLIOGRAPHIC AUTOMATION OF A LARGE LIBRARY USING A TIME-SHARING SYSTEM. COMPUTERS AND HUMANITIES,VOL 5,NO.5,5/71
OWNER:
CALHOUN

DOCUMENT NO. 5
SHAMIR, E.
SOME INHERENT AMBIGUITY OF CERTAIN CONTEXT-FREE LANGUAGES, DEP. OF MATH.,HEBREW U, JERUSALEM,ISRAEL,INFOR AND CONTROL,VOL. 18,NO. 4, MAY 1971, PP.355-363
OWNER:
HANKLY

DOCUMENT NO. 6
FLECK, A.
ON THE COMBINATORIAL COMPLEXITY OF CONTEXT-FREE GRAMMARS.
IN PROC. OF THE IFIP CONGRES 71, '72,59-60
OWNER:
WALLENTINE

DOCUMENT NO. 8
ELLIS, C.
THE HALTING PROBLEM FOR PROBABILISTIC CONTEXT-FREE GENERATORS.
J. ACM. 19,(JULY 1972)396-399
OWNER:
FISHER

DOCUMENT NO. 9
AHO, A.
DENINING, P.
WEAK AND MIXED STRATEGY PRECEDENCE PARSING.,J. ACM 19,2(4/72)
OWNER:
FISHER

DOCUMENT NO. 11
HOPCROFT, J.
ULLMAN, J.
FORMAL LANGUAGES AND THEIR RELATION TO AUTOMATA.,ADISON-WESLEY.
OWNER:
HANKLY

ORDKWS

LIST OF KEY WORDS AND DOCUMENT NUMBER(S) CROSS REFERENCES:

NO. 3: *AUTOMATA* ----------11
NO. 12: *CONTEXT-FREE GRAMMARS* ----------6
NO. 11: *CONTEXT-FREE LANGUAGES* ----------5
NO. 15: *DATA TRANSMISSION* ----------2
NO. 23: *FORMAL LANGUAGES* ----------5 6 11 9
NO. 35: *INFORMATION RETRIEVAL* ----------4
NO. 47: *LIBRARIES AND INFORMATION CENTER* ----------4
NO. 51: *MIXED STRATEGY PRECEDENCE PARSING* ----------9
NO. 64: *PROBABILISTIC CONTEXT-FREE GENERATORS* ----------8
NO. 63: *PROBABILISTIC GRAMMARS* ----------8
NO. 79: *SYSTEM ANALYSIS* ----------2
NO. 95: *WEAK PRECEDENCE PARSING* ----------9

```
      )COPY FNS CREATION1
SAVED  19.45.32 04/10/74

      ENTER
ENTER ONE LINE PLEASE
A_FLECK, A.
ENTER ONE LINE PLEASE
BREWER
ERROR:INPUT PROCEDURE WAS IMPROPER, CORRECT THE ABOVE LINE BY RET
      YPING THE RIGHT STRING
ENTER ONE LINE PLEASE
O_BREWER
ENTER ONE LINE PLEASE
T_TOWARD A THEORY OF DATA STRUCTURES, J. COMPUTER AND SYSTEMS SCIENCE
TEXT INFORMATION OVER 65 CHARACTERS,TYPE AGAIN
T_TOWARD A THEORY OF DATA STU
                             v
                             RUCTURES, J. COMPUTER AND SYSTEMS
ENTER ONE LINE PLEASE
T_SCIENCE, OCT. '71, 475+
                          v
                          -488
ENTER ONE LINE PLEASE
K_CONTEXT-FREE LANGUAGES
ENTER ONE LINE PLEASE
K_DATA STRUCTURES
ENTER ONE LINE PLEASE
K_NET WORK GRAMMARS
ENTER ONE LINE PLEASE
END
THE DATA WERE SUCCESSFULLY ENTERED,IF YOU WANT TO CHECK TYPING ER
      ROR TYPE DISPLAY,OTHERWEISE TYPE STORE

      STORE
DOCUMENT NO. 12
AUTHOR'S NAME HAS BEEN STORED
TEXT INFORMATION HAS BEEN STORED
NAME OF OWNER HAS BEEN STORED
ENTER:)SAVE W3
      )LOAD W4
      )COPY W3 KWS
      ADKEYS
      )SAVE W3
  16.48.52 05/15/74
      )LOAD W4
SAVED  16.17.31 05/15/74
      )COPY W3 KWS
SAVED  16.48.52 05/15/74
      ADKWYS
         v
         EYS
VALUE ERROR
      ADKEYS
      ^
      )COPY FNS CREATION2
SAVED  19.45.32 04/10/74
```

```
      ADKEYS
KEY WORDS HAVE BEEN STORED

ENTER: )SAVE W4
              FOR ENTERING ANOTHER DOCUMENT TYPE:
    1:)LOAD W3
    2: ENTER

      )SAVE W4
  16.51.17 05/15/74
```

```
    )LOAD W3
SAVED  16.48.52 05/15/74
       DELETE
ENTER TO BE DELETED DOCUMENT:
□:
       8
AUTHOR(S) DELETED
ENTER:
     1:)SAVE W3
     2:)LOAD W4
     3: )COPY FNS DTKEY (AS NEEDED)
     4: )COPY W3 DTNO
     5: )DTKEY

       DELETE
ENTER TO BE DELETED DOCUMENT:
□:
       9
AUTHOR(S) DELETED
ENTER:
     1:)SAVE W3
     2:)LOAD W4
     3: )COPY FNS DTKEY (AS NEEDED)
     4: )COPY W3 DTNO
     5: )DTKEY
       )SAVE W3
  19.36.52 05/15/74
       )LOAD W4
SAVED  16.51.17 05/15/74
       DTNO←8
       DTKEY
KEY WORD(S) DELETED
ENTER: )SAVE W4
       DTNO←9
       DTKEY
KEY WORD(S) DELETED
ENTER: )SAVE W4
       )SAVE W4
  19.38.25 05/15/74


       )LOAD W3
SAVED  19.36.52 05/15/74
       ENTER
ENTER ONE LINE PLEASE
A_SALTON, G.
ENTER ONE LINE PLEASE
T_RECENT STUDIES IN AUTOMATIC TEXT ANALYSIS AND DOCUMENT RETRIEVAL
ENTER ONE LINE PLEASE
T_J. ACM, VOL 20,NO. 2, APRIL '73, PP. 258-278
ENTER ONE LINE PLEASE
K_AUTOMATIC TEXT PROCESSING
ENTER ONE LINE PLEASE
K_INFORMATION RETRIEVAL
ENTER ONE LINE PLEASE
```

```
K_AUTOMATIC INDEXING
ENTER ONE LINE PLEASE
K_COMPUTATIONAL LINGUISTICS
ENTER ONE LINE PLEASE
O_HANKLY
ENTER ONE LINE PLEASE
END
THE DATA WERE SUCCESSFULLY ENTERED,IF YOU WANT TO CHECK TYPING ER
      ROR TYPE DISPLAY,OTHERWEISE TYPE STORE


      )COPY FNS COLLECT
SAVED  19.45.32 04/10/74

      COLLECT
DOCUMENT NO.3
TEXT INFORMATION HAS BEEN STORED
AUTHOR'S NAME HAS BEEN STORED
NAME OF OWNER HAS BEEN STORED
ENTER:
    1:)SAVE W3
    2:)LOAD W4
    3:)COPY W3 KWS
    4: ADKEYS

      )SAVE W3
  19.49.18 05/15/74
      )LOAD W4
SAVED  19.38.25 05/15/74
      )COPY W3 KWS
SAVED  19.49.18 05/15/74
      ADKEYS
KEY WORDS HAVE BEEN STORED

ENTER: )SAVE W4
            FOR ENTERING ANOTHER DOCUMENT TYPE:
    1:)LOAD W3
    2: ENTER
      )SAVE W4
  19.50.22 05/15/74
```

```
      )LOAD W3
SAVED  19.49.18 05/15/74


      SQNC

DOCUMENT NO. 2
MARTIN, J.
SYSTEMS ANALYSIS FOR DATA TRANSMISSION. PRENTICE-HALL,
ENGLEWOOD CLIFFS, N. J., 1972, 909 PP.
OWNER:
BREWER

DOCUMENT NO. 3
SALTON, G.
RECENT STUDIES IN AUTOMATIC TEXT ANALYSIS AND DOCUMENT RETRIEVAL
J. ACM, VOL 20,NO. 2, APRIL '73, PP. 258-278
OWNER:
HANKLY

DOCUMENT NO. 4
VEANER, A.
BALLOTS-BIBLIOGRAPHIC AUTOMATION OF A LARGE LIBRARY USING A
TIME-SHARING SYSTEM. COMPUTERS AND HUMANITIES,VOL 5,NO.5,5/71
OWNER:
CALHOUN

DOCUMENT NO. 5
SHAMIR, E.
SOME INHERENT AMBIGUITY OF CERTAIN CONTEXT-FREE LANGUAGES,
DEP. OF MATH.,HEBREW U, JERUSALEM,ISRAEL,INFOR AND CONTROL,VOL.
18,NO. 4, MAY 1971, PP.355-363
OWNER:
HANKLY

DOCUMENT NO. 6
FLECK, A.
ON THE COMBINATORIAL COMPLEXITY OF CONTEXT-FREE GRAMMARS.
IN PROC. OF THE IFIP CONGRES 71, '72,59-60
OWNER:
WALLENTINE

DOCUMENT NO. 11
HOPCROFT, J.
ULLMAN, J.
FORMAL LANGUAGES AND THEIR RELATION TO AUTOMATA.,ADISON-WESLEY.
OWNER:
HANKLY

DOCUMENT NO. 12
FLECK, A.
TOWARD A THEORY OF DATA STRUCTURES, J. COMPUTER AND SYSTEMS
SCIENCE, OCT. '71, 475-488
OWNER:
BREWER
```

```
      )COPY FNS DUMP2
SAVED  19.45.32 04/10/74

      ORDKWS
```

LIST OF KEY WORDS AND DOCUMENT NUMBER(S) CROSS REFERENCES:

NO. 3: *AUTOMATA*	----------11
NO. 5: *AUTOMATIC INDEXING*	----------3
NO. 4: *AUTOMATIC TEXT PROCESSING*	----------3
NO. 17: *COMPUTATIONAL LINGUISTICS*	----------3
NO. 16: *CONTEXT-FREE GRAMMARS*	----------6
NO. 15: *CONTEXT-FREE LANGUAGES*	----------5 12
NO. 24: *DATA STRUCTURES*	----------12
NO. 23: *DATA TRANSMISSION*	----------2
NO. 31: *FORMAL LANGUAGES*	----------5 6 11
NO. 43: *INFORMATION RETRIEVAL*	----------3 4
NO. 55: *LIBRARIES AND INFORMATION CENTER*	----------4
NO. 63: *NET WORK GRAMMARS*	----------12
NO. 87: *SYSTEM ANALYSIS*	----------2

```
      )LOAD W1
SAVED  16.12.00 04/07/74

      BYAUTHOR
ENTER  NAME:
SALTON, G.
SALTON, G.       IS NOT IN W1
ENTER  NAME:

      )COPY F
            v
            W3 DATA1
SAVED  19.49.18 05/15/74

      BU
      v

      BYAUTHOR
ENTER  NAME:
SALTON, G.

DOCUMENT NO. 3
SALTON, G.
RECENT STUDIES IN AUTOMATIC TEXT ANALYSIS AND DOCUMENT RETRIEVAL
J. ACM, VOL 20,NO. 2, APRIL '73, PP. 258-278
OWNER:
HANKLY
ENTER  NAME:




      )LOAD W2
SAVED  13.24.13 04/05/74

      BYKEYWORD
ENTER KEY WORD(S):
AUTOMATIC INDEXING
AUTOMATIC INDEXING                              IS NOT IN W2
ENTER KEY WORD(S):

      )COPY FNS
      v
      )COPY W3
      v
      )COPY W4 DATA2
SAVED  19.50.22 05/15/74
```

```
      BYKEYWORD
ENTER KEY WORD(S):
AUTOMATIC INDEXING
KEY WORDS FOUND TYPE:
     1:)SAVE W4
     2:)LOAD W3
     3:)COPY FNS PRINTOUT
     4:)COPY W4 NLIST
     5: PRINTOUT

      )SAVE W4
NOT SAVED, THIS WS IS W2

      )COPY W3 DATA1
SAVED  19.49.18 05/15/74

      PRINTOUT

DOCUMENT NO. 3
SALTON, G.
RECENT STUDIES IN AUTOMATIC TEXT ANALYSIS AND DOCUMENT RETRIEVAL
J. ACM, VOL 20,NO. 2, APRIL '73, PP. 258-278
OWNER:
HANKLY


      )LOAD W3
             v
             4
SAVED  19.50.22 05/15/74

      BYKEYWORD
ENTER KEY WORD(S):
FORMAL LANGUAGES
KEY WORDS FOUND TYPE:
     1:)SAVE W4
     2:)LOAD W3
     3:)COPY FNS PRINTOUT
     4:)COPY W4 NLIST
     5: PRINTOUT
      )SAVE W4
  20.31.10 05/15/74
      )LOAD W3
SAVED  19.49.18 05/15/74

      )COPY FNS PRINTOUT
SAVED  19.45.32 04/10/74
      )COPY W4 NLIST
SAVED  20.31.10 05/15/74
```

PRINTOUT

DOCUMENT NO. 5
SHAMIR, E.
SOME INHERENT AMBIGUITY OF CERTAIN CONTEXT-FREE LANGUAGES,
DEP. OF MATH.,HEBREW U, JERUSALEM,ISRAEL,INFOR AND CONTROL,VOL.
18,NO. 4, MAY 1971, PP.355-363
OWNER:
HANKLY

DOCUMENT NO. 6
FLECK, A.
ON THE COMBINATORIAL COMPLEXITY OF CONTEXT-FREE GRAMMARS.
IN PROC. OF THE IFIP CONGRES 71, '72,59-60
OWNER:
WALLENTINE

DOCUMENT NO. 11
HOPCROFT, J.
ULLMAN, J.
FORMAL LANGUAGES AND THEIR RELATION TO AUTOMATA.,ADISON-WESLEY.
OWNER:
HANKLY

ACKNOWLEDGEMENTS

The author is deeply indebted to his major professor, Dr. Willam J. Hankly, for his advice and guidance in the preparation of this report. The author also wishes to express his sincere appreciation to his committee members, Dr. Richard K. Brewer and Dr. Myron A. Calhoun, for their corrections of the manuscript and criticisms on the work. In addition, thanks are extended to the Department of Computer Science for providing computing time.

AN INTERACTIVE LITERATURE COLLECTION AND REFERENCE RETRIEVAL SYSTEM

by

YU-JEN EUGENE TSENG

B.A., NATIONAL CHENGCHI UNIVERSITY, 1966

AN ABSTRACT OF A MASTER'S REPORT

submitted in partial fulfillment of the

requirements for the degree

MASTER OF SCIENCE

Department of Computer Science

KANSAS STATE UNIVERSITY
Manhattan, Kansas

1974

ABSTRACT

An interactive information retrieval system was built as a pedagogical exercise, with the objective of demonstrating the organization of total system. The system stores names of documents together with authors, reference information, keywords, owner information, and reading lists for subject areas. The main purpose of maintaining this system is to provide reference retrieval by author's name, name of document, keyword, key phrase, or document number. It is useful in cases where fast response is needed for small volumes of retrieved data. General system description and technical overview are in Chapters I, and II. A user's guide is in Chapter III. The rest of this report provides technical details of the system.