This double project is submitted in partial fulfillment for the M.Sc.A degree in the School of Computer Science, McGill University

Hua-i Chiu Franks
7819156
September,1986

BIBLIOGRAF

A SYSTEM FOR STORAGE AND RETRIEVAL
OF BIBLIOGRAPHIC REFERENCES ON A MICROCOMPUTER

Table of Contents

I. INTRODUCTION

1. Scope

Over the past few years the technological revolution in hardware combined with a reduction in the cost of computer processing power has led to an increase in the use of computers as an aid to biomedical research. The advantages of using a computer include speed, accuracy and flexibility.

One of the areas in biomedical research which requires that information be accurate and readily accessible involves the classification of bibliographic reference materials. Management of scientific literature references is an extremely important but very time-consuming task for the scientist. There are programs that exist for managing a bibliographic reference storage system, but most have been designed for medium to large computers, not desktop models.

2. Existing System

To date most scientists have relied on a manual indexing system for classification of their bibliographic references. A typical index system might include a small metal box which would contain hand-posted or typed 3x5 cards arranged alphabetically or numerically. Another system might include a card catalog file similar to a library classification scheme, with papers arranged by author or subject. The cards would contain some information for each reference such

as authors, year of publication and perhaps a few descriptive keywords. Information would be added to a card as it enters the system while details would be altered or deleted as needed. However, as time passes and the file grows larger and more complex, the system would probably contain many duplicate records for a particular reference, particularly if a cross-referencing system were to be established at a later date.

It is obvious, then, that a manual system for storing bibliographic references would become extremely cumbersome as more references were added to it, and would prove to be very unwieldy particularly when more than a few references needed to be retrieved. Since a vast majority of laboratories now have access to desktop computers both for data acquisition and analysis, a computerized system to store bibliographic references would be a definite asset within the research environment. Such a system would meet the requirements that data be consistent,available when required, shared among laboratory personnel and managed efficiently and effectively.

3.Proposed System

In order to overcome many of the problems that exist with the present manual system, i.e., factors such as

-accuracy of information within the system

-efficiency of processing

-scope of output reports

the proposed system would include the following capabilities:

-entry/edit function to enter new information or correct stored information

-retrieval function to search files to obtain a listing of documents by file number

-print function to printout information stored for each document.

Since a computerized system would lend itself to fast access to the data and is designed for string manipulation, the basis of the system would be classification of bibliographic references by keyword. The scientist himself would then control what documents were to be included in the reference database. Keyword assignment,moreover, would have a major impact on how complete and accurate the subject matter was represented in the reference system.

Another factor of major importance in a computerized reference system would involve the subsystem of interaction between user and the system, that is, the user-system interface. Since the researcher or other laboratory personnel may not be competent in a programming language or lack funds to hire a skilled programmer, what is needed is a system which would allow data storage and retrieval to be handled by personnel not having a high level of computing expertise. This user-oriented software system would consist

of a series of interactive programs which would facilitate data input and manipulation as well as data output. This objective would be accomplished by a menu-based subsystem which would combine information about authors or Keywords in an index file that would serve as a pointer to a master file on which information for the complete reference is stored. Thus the system would prompt the user to enter information pertaining to authors, title, journal, volume number and year of publication followed by significant keywords that would give some idea as to the contents of the reference article.

Hence factors such as how good the user-system interaction is and how well the searching strategy performs will have a significant influence on output from the system. Since the system is one based on author or keyword, the researcher who is performing a search must break down his search request into categories or subsystems such as organs or disease in order to scan references from very broad topics to very specific ones.

II. DESCRIPTION OF SYSTEM

1. Storage Algorithm

The storage algorithm on which this system is based involves the use of hashing which is a technique for accessing direct access or random access files. In general,

hashing utilizes a primary key K and involves an arithmetical calculation for some part of that key that results in a function f(K) which gives the location of K in an index table where the storage or search of K begins. However, there is a possibility that two distinct keys K(i) = K(j) may hash to the same location, resulting in a collision. If a collision occurs, then it must be resolved by a collision resolution method. One such CRM is chaining in which numerous linked lists are maintained, one emanating from each individual hash address. Furthermore each bucket that handles the overflow records from a hash address may in turn have its own separate chain of overflow records. The chaining method is generally fast because most of the lists of overflow records tend to be short. For example, if there are N keys and M lists, then the average list size is N/M.

Another collision resolution method is open addressing which involves looking at each location in the index table to find either a match with the key K or an empty position where the key can then be stored. If the latter is done, then appropriate steps to update the chain from the original hash location must be made. The simplest scheme of the open-addressing technique is that of linear probing, using the cyclic probe sequence

$$h(K), h(K)-1, \ldots 0, (m-1), (m-2), \ldots, h(K)+1.$$

The disadvantage of this method, however, is that when the table starts to get full, the need to perform linear probing

becomes more frequent and the time required to accomplish this procedure increases drastically.

The method chosen as the basis of storage for this bibliographic system is a further refinement of the concept of hashing. The technique, as proposed by its author (Litwin, 1980), is known as linear virtual hashing and is based on the concept that the file or address space grows or shrinks dynamically by modifying the hash function during the course of insertions or deletions. This dynamic method of storing records into blocks on a disk file is achieved by utilizing a sequence of related hash functions that allows a block to be split when a collision occurs and its contents reallocated to other blocks depending on the new addresses computed by using the next hash function in the sequence.

Thus, if the hash function sequence is:

$h_0(K)=K \bmod N$ where N is the number of blocks originally in the file,

then the sequence of hash functions is generated by

$h_i(K)=K \bmod 2^i N$ where $h_i, i=0,1,2,\ldots$ is a uniform hash function.

Furthermore, if a collision does occur and more storage is required, then it is allocated only one block at a time. Since the block that is split is usually not the one where the collision has occurred, then a pointer P must be used to

keep track of the next block to be split. As a result, reorganization of the file occurs in a predefined order, i.e., 0,1,2,3,...(N-1) and the address space increases linearly until it has doubled. Thus the amount of storage available for each hash function is twice that of the previous function.

Another factor which influences this file reorganization involves the existence of a control level which determines when a block is to be split. This load control factor is defined as follows:

$\alpha = x/b*m$ where x=no of file records
b=bucket capacity
m=number of primary buckets.

and is calculated whenever a collision occurs. The resulting factor is compared to a threshhold level, α_T, which is initially set at 0.8. If $\alpha > \alpha_T$, then a bucket is split and the file is subsequently reorganized. In this way, storage utilization is kept constant according to a level designated by the user and an accumulation of overflow records from a particular bucket is avoided.

Description of algorithm

1.Initial conditions:

Jptr - ptr to the current location in the hash function sequence

- hash address H2 is computed using Jptr

- hash address H1 is computed using (Jptr-1), ptr to the previous location in the hash function sequence.

P - ptr to the next block to be split

Talpha - load control factor = 0.8

Nrec - no. of records stored = 0

Nb - no. of blocks allocated = 1

$h_{-1}(K)=-1$ for any key K

2.Input key K:

3.Hash:

Nrec=Nrec + 1

H(Jptr)←Nb

N←H(Jptr)

Hash address computation:

AK mod w where w=2^16

$$A=(\sqrt{5}-1)* w/2 \text{ (Knuth,1973)}$$

Compute Adr →H2

Test: if Jptr>0 then N←H(Jptr-1); compute Adr←H1

Test:if P=0 then Adr=H2 else if H1<P then Adr=H2 else Adr=H1.

- if block is not full, store key at address Adr and get

next key.

4.Split Control:

Compute Alpha=Nrec/(Nb*blocksize)

Test: if Alpha <= Talpha then store the record as an overflow from block=Adr and update chain of pointers from last record in block Adr.

Else allocate new block:

Test: if P=0 then Jptr=Jptr+1

Calculate Nublck = P + (2^(Jptr-1)*v)

Increment Nb=Nb+1

Calculate Adr and store record as overflow from Adr.

Update block pointers.

5.Split:

Prepare for rehashing records in block P and its overflows:

Check for foreigner records in P:

-calculate Adr using H(Jptr-1)

-test: if Adr=P then push key on stack for rehashing else save foreigner records in separate array

Initialize records in P and overflows

Rehash records in P including its overflows using H(Jptr)

Reinsert foreigner records and update block pointers

6.Update control pointers:

-increment P=P+1

-test: if P>=(2^(Jptr-1)*v) then P=0.

7.Get next key.

Thus the scheme for linear hashing uses the technique which addresses records based on a primary key. Records are inserted into a specific bucket until the latter is full, a collision occurs, and a CRM comes into effect. If the load control factor is greater than a predetermined threshhold level, then a new hash function is computed for all records at a location designated by a pointer P with the result that some of these records will be assigned to other buckets depending on the new addresses calculated from this reorganization.

An illustration of linear virtual hashing is presented below.

Initial values:
P=0, J=0, Blocksize=3, V=1, Talpha=0.8, H (K)=-1 for any K

Input:	Adr	Blk	Rec	P	J
1.D.Franks	0	0	1	0	0
P.Hamet	0	0	2	0	0
J.Plamondon	0	0	3	0	0
H.Chiu	0	collision alpha=1.33,nublck		0	1

Rehash:				1	1
After rehash:				0	1
H.Chiu	0	0	1		
P.Hamet	0	0	2		

J.Plamondon	1	1	4		
D.Franks	1	1	5		

2.Input:					
D.Franks	1	1	6	0	1
H.Chiu	0	0	3	0	1
J.Doe	0	collision		0	2
		alpha=1.16,nublck			
Rehash:				1	2
After rehash:				1	2
J.Doe	0	0	1		
H.Chiu	0	0	2		
P.Hamet	0	0	3		
J.Plamondon	1	1	4		
D.Franks	1	1	5		
D.Franks	1	1	6		
H.Chiu	0	2	7		
Input:	Adr	Blk	Rec	P	J
S.Kirby	1	collision		1	2
		alpha=0.88,nublck			
Rehash:				2	2
After rehash:				0	2
J.Doe	0	0	1		
H.Chiu	0	0	2		
P.Hamet	0	0	3		
D.Franks	2	1	4		

H.Chiu	2	2	7		
S.Kirby	2	2	8		
D.Franks	2	2	9		
J.Plamondon	3	3	10		

3.Input					
P.Hamet	0	collision		0	2
		alpha=0.75,store			

```
After store:
  P.Hamet          0     3     11          0     2
Input
  J.Plamondon      3     3     12          0     2
  D.Franks         2     collision         0     3
                         alpha=0.91,nublck
Rehash:                                    1     3
After rehash:                              1     3
  P.Hamet          0     0     1
  H.Chiu           0     0     2
  P.Hamet          0     0     3
  D.Franks         2     1     4
  J.Doe            1     1     5
  -----
  -----
  S.Kirby          2     2     8
  D.Franks         2     2     9
  J.Plamondon      3     3     10
  -----
  J.Plamondon      3     3     12
  D.Franks         2     4     13
  H.Chiu           0     4     14
```

2.Hardware Configuration

This system has been implemented on a 48K Apple II Plus microcomputer containing a 16K language card and the Applesoft II Basic language in ROM. In addition there are two floppy disk units attached which allow the user to store and retrieve information from files that are stored on the 5 1/4 inch diskettes. Each diskette contains 35 tracks for recording information; these are further divided into 16 sectors each of which holds 256 bytes of information. Thus the total storage space allotted to the user is approximately 122K bytes per diskette, having made an allowance for housekeeping information such as file directory to fill the remaining storage space. The resident software program that handles the housekeeping tasks such as keeping track of files on the diskette and saving and retrieving information is called the Disk Operating System or DOS and is version 3.3. The last piece of equipment includes the Apple dot matrix printer which outputs 80 characters per second and requires a parallel interface card to operate.

3.Software Components

The present system consists of a series of programs which allow the user to (1) initialize or format a blank diskette, (2) back-up or copy a file, (3) initialize an index file for author, keyword or master file when the system is first

initiated and (4) store and retrieve a bibliographic reference from the master file. All programs have been written in the Applesoft Basic programming language. The programs to initialize blank diskettes or copy a file have been incorporated from the Apple II Plus software library. The others have been written specifically to allow the user to store and retrieve a bibliographic reference. The main source program includes a number of menu displays and subroutines which allow the user to input data or retrieve and print a reference listing interactively. A description of the software components follows.

A.Menu Displays

1.Menu 1 - is a listing of user options available within the bibliographic system. Each option represents a major function of the system and the user will always be returned to this menu for alternate routing or to quit the system. The help option gives a general description of the overall reference system.

Help	H
Enter reference	E
Search reference	S
Quit the system	Q

2.Menu 2 - is a listing of user options after the Enter reference option has been selected from Menu 1.

Help	H
Input	I
Check	C
Store	S
Quit	Q

Help - this facility serves as a guide to entering the reference.

Input - is the option selected to enter a new reference into the system.

Check - is the option that provides the user with the ability to correct the data that he has just finished inputting. Since the full references are stored in a variable length format on the main reference file, it would be difficult to correct one reference without altering every reference thereafter. Thus the user must use the Check facility to insure that the data is accurate before it enters the system.

Store - is the system function that stores a reference. This process involves storing the authors in the author index file, the keywords in the keyword index file and the full reference in the master reference file.

Quit - allows the user to return to the Menu 1 for alternate routing or to quit the system entirely.

3.Menu 3 - is a listing of user options after the Search option has been selected from Menu 1.

Help	H
Authors	A
Keywords	K
Printout	P
Quit	Q

Help - this option serves as a guide to search the reference file for specific articles based on author or Keyword.

Authors - prompts the user to enter author names as the basis of a search.

Keywords - prompts the user to enter descriptive Keywords as the basis of a search.

Printout - allows the user to print the contents of a temporary file that holds the results of a successful search. The system will display Menu 4 which presents the user with various printout options.

4.Menu 4 - is a listing of print instructions to help the user adapt the printout to suit a particular need.

As found	F
User ordered	U
Quit	Q

As found - references are printed out in the order in which they were located in the files.

User ordered - references are printed out in a particular order, as determined by the user's needs.

Quit - returns the user to Menu 3.

5.Menu 5 - allows the user some options for additional searches or printouts.

New file	N
Add to file	A
Alternate printout	L
Quit	Q

New - clears the temporary file of past 'hits', thus insuring that the next search will be a new one.

Add - allows the user to retain the 'hits' from previous searches and merges them with the results of the next search.

Alternate printout - similar to the 'user ordered' option in Menu 4.

Quit - returns the user to Menu 3.

B.Programs

1.Crindx - allows the user to create a sequential data file with information on pointers that will be used in the author index, keyword index and main reference files. This program should only be run at the initiation of a new Biblio system or else existing pointers to records in the files will be erased.

2.Initfl - allows the user to initialize a data file for storing index or reference information. Essentially this program creates a file with 300 records and initializes it to receive alphanumeric data.

3.Biblio - is the main program for storage and retrieval of a reference. When the user initiates a start-up procedure by turning the power on after inserting the System master diskette into Drive 1, he will be presented with a display of the system logo and then immediately thereafter with the main menu. At this point, the user has the option of continuing to interact with the system or to terminate the session. If the user continues on the system he will then

opt to enter a reference or search for a reference. In either case he will be presented with another appropriate menu from which he may proceed by selecting more specific options. The user is also able to obtain assistance for any aspect of the system with which he is unfamiliar by consulting an on-line help facility under the appropriate system functions. Thus during the course of using the system, the user will be guided at all times by system prompts or menus from which to continue processing. When the user wishes to terminate a session, certain file pointers that describe the state of the files must be stored back on specific diskettes. In this case the system will prompt the user to insert the appropriate diskettes. When all this housekeeping has been done, the session will end.

C.Subroutines of Biblio

1.Setln (line 30) - sets up a formatted alphanumeric string of specific length to be written to a record on the disk.

2.Comphshno (line 90) - computes a hash number as an entry point into either the author or keyword index file. The arithmetical computation is as follows:

Akmod w where w=2^16

A=(√5-1)* w/2 (Knuth,1973)

3.Dskin (line 1970) - checks to see if the disk already in drive 1 is the correct disk. At this stage of processing, the information is read from the appropriate file, record 0. If the disk volume is correct, processing continues or else the user is advised of a volume mismatch and is given the opportunity to insert the correct disk.

4.Cktbl (line 2120) - scans the records in a specific bucket of an index file pointed to by the computed hash number to see if storage space is available. If it is, then the empty record number is returned to the supervisor program for further processing. If the bucket is full, then a flag is set to indicate a collision has occurred and a CRM must follow.

5.Linprob (line 2330) - is used to calculate an address different from the original hash address. This routine is based on the cyclic probe sequence that was previously presented.

6.Storeht (line 2380) - formats the string containing the author or keyword name, volume (disk) number, record number and overflow pointer prior to storage on a record in the appropriate index file.

7.Updte (line 2460) - when space for a new record to be stored has been found, this subroutine updates the pointer information on the last record in the chain of overflow records emanating from the record's block address.

8.Prfile (line 2730) - prints out the contents of a file from block 0 through the last block address as indicated by the user.

9.Help (line 2840) - extracts text from a file and displays it on the screen.

10.Push (line 3000) - increments a stack counter and pushes the name (author or keyword) onto a stack for further processing during the reorganization of the file.

11.Getkw (line 3050) - extracts the author or keyword name from the field allocated for that purpose on each block record.

12.Getlstrec (line 3130) - follows the chain of overflow records from a block address. The pointer is stored in the last record of each block address.

13.Vrfydsk (line 3230) - verifies that the disk in a specific drive matches the one required for processing. If there is no match, an error message is printed and the user is requested to insert another disk. The verify procedure is then repeated.

14.Lvhi (line 3430) - Linear virtual hash insert is the main routine that processes the input reference data. The sequence of events is as follows. Once a name has been entered, a hash number is calculated for it. This number serves as the pointer to a block address in the appropriate index file. The file is checked to see if there is storage space available. If so, the input reference data is processed. If the specific block is full, then an overflow procedure is initiated. Depending on the load factor, either the new data is just stored at another block address or file reorganization takes place after new hash functions are computed for those records in the block indicated by the pointer p as the next block to be split.

15.Getchar (line 6050) - extracts the information from a string and stores it into a new string.

16.Kwserch (line 6130) - once an author's name or specific keyword has been input as the basis of a search, it is converted to a hash number which is the address where the search will begin. If a match occurs, the location on the master reference file is stored on a temporary file for further processing. The same information is also stored for any matches that may have occurred after the chain of records is followed from the original block address.

III. DISCUSSION

1. Operator interface

In order that the system be used with minimal user effort and training, all the specific functions that the user chooses, such as entering a reference or printing out the master file, are performed under the guidance of the system. In addition, in order to maintain a user-friendly environment, menus are used in the presentation of specific options once a function has been selected. The element of choice, as demonstrated by the routing alternatives available within the system , reinforces the idea that the user is in control of the system at all times.

Thus the new system that has been designed to facilitate storage and retrieval of bibliographic references for the laboratory scientist should be advantageous in several ways. Hopefully once the system is installed, it should help to reduce manpower time and long term operating costs to maintain a literature classification scheme by decreasing the processing time for storage or retrieval of a reference article and by allowing for more flexible outputs. The latter is important because most of the scientific journals have differing format requirements for their bibliographic listings. With features such as menu displays, single key responses and the availability of a self-help key, hopefully

entry errors would be minimized and the user would soon gain a familiarity with the system and feel comfortable working in a computerized environment. The advantages of this system over a manual system including increase in the size of the database, increased accessibility and increased flexibiliity of output formats would outweigh the disadvantages of the amount of disk shuffling that would be required if the database expanded rapidly. If the overwhelming constraint is volume, then the scientist could perhaps divide a large number of references into several smaller databases depending on year of publication.

2.Conclusion

This project, then, is an initial attempt to design and implement a bibliographic reference system to be used in a research laboratory. This system is not intended to be a comprehensive reference system due to space constraints. Frequent housekeeping will be necessary to keep only the most recent references in the active files.

BIBLIOGRAPHY

REFERENCES

Knuth, D.E., The Art of Computer Programming, Vol.III, Addison-Wesley, Reading, Mass., 1973.

Litwin, W., Linear Hashing: A New Tool for File and Table Addressing, Proceedings of the Sixth International Conference on Very Large Databases, Montreal, Canada, Oct.1-3, 1980, p.212-23.

PROGRAM LISTINGS

```
10  REM  CRINDX
20  DATA  0001,0000,0001,0000,000
     0,0000,0001,0000,0001,0000,0
     000,0000,0001,01,001
30 D$ =  CHR$ (4):R$ =  CHR$ (13)
     : DIM DTA$(15)
40  FOR J = 1 TO 15
50  READ DD$:DTA$(J) = DD$
60  NEXT J
80  HOME
90 RSP$ = "": VTAB 5: INPUT "IS T
     HIS A NEW FILE?(Y/N)";RSP$
100  IF RSP$ = "Y" THEN 180
110 RSP$ = "": VTAB 10: INPUT "CA
     UTION-YOUR FILE OF PTRS TO T
     HE AUTHOR,KEYWORD AND MASTER
      REFERENCE FILES IS ABOUT TO
      BE DELETED,THEN REINITIALIZ
     ED -- (Y/N)?";RSP$
120  IF RSP$ = "Y" THEN 160
130  IF RSP$ = "N" THEN 500
140  GOTO 110
160  PRINT D$;"OPEN INDXPTRS,L31"

170  PRINT D$;"DELETE INDXPTRS"
180  PRINT D$;"OPEN INDXPTRS,L31"

190 IREC = 0:ST$ = ""
200 STRT% = 1:ND% = 6
210  FOR I = STRT% TO ND%
220 ST$ = ST$ + DTA$(I)
230  NEXT I
240  PRINT D$;"WRITE INDXPTRS,R";
     IREC
250  PRINT ST$
260 IREC = IREC + 1:ST$ = ""
270  IF IREC > 2 THEN 400
280 STRT% = ND% + 1:ND% = STRT% +
     5
290  IF ND% > 15 THEN ND% = 15
300  GOTO 210
400  PRINT D$;"CLOSE INDXPTRS"
450  PRINT "FILE INITIALIZED"
500  END

]
```

```
 LOAD INITFL
]LIST

10  REM  ***INITFL***
20  CLEAR
30 D$ =  CHR$ (4):R$ =  CHR$ (13)

40 MDSKS$ = "0001":NREC$ = "0000"
     :NB$ = "0001":P$ = "0000":JP
     TR$ = "0000":TMP$ = "0000"
50 VL$ = "01":BNO$ = "001"
60 KW$ = "00000000000000000000"
70 VOL$ = "00"
80 BLCK$ = "000"
90 OVR$ = "0000"
100  HOME : INPUT "ENTER FILETYPE
      - (0/1/2):";TYP%
110  IF TYP% = 0 THEN  GOTO 570
120  ON TYP% GOTO 130,470
130  REM   INITIALIZE AWTBL OR KW
     TBL:HOME
140  INPUT "ENTER FLNME,RECLEN,DR
     VNO:";FLNME$,RLN,DV$
150  INPUT "ENTER NO RECORDS:";NM
     B%
160  PRINT "INSERT DISK INTO DRIV
     E ";DV$;"THEN PRESS <RETURN>
     "
170  GET CH$
180  IF CH$ = R$ THEN  GOTO 190: GOTO
     170
190  PRINT R$
200  PRINT D$;"OPEN";FLNME$;",L";
     RLN;",D";DV$
210  PRINT D$;"DELETE";FLNME$
220  PRINT D$;"OPEN";FLNME$;",L";
     RLN;",D";DV$
230 ST$ = MDSKS$ + NREC$ + NB$ +
     P$ + JPTR$ + TMP$
240  PRINT D$;"WRITE";FLNME$;",R0
     "
250  PRINT ST$:ST$ = ""
260 ST$ = KW$ + VOL$ + BLCK$ + OV
     R$
270  FOR I = 1 TO NMB%
280  PRINT D$;"WRITE";FLNME$;",R"
     ;I
290  PRINT ST$
300  NEXT I
310  PRINT D$;"CLOSE";FLNME$
320  INPUT "DO YOU WANT A LISTING
     ?(Y/N)";CH$
330  IF CH$ = "Y" THEN 350
340  GOTO 100
350  PRINT D$;"OPEN";FLNME$;",L";
     RLN
360  INPUT "ENTER STRT,END:";STRT
     ,ND
370  FOR J = STRT TO ND
380  ONERR  GOTO 440
390  PRINT D$;"READ";FLNME$;",R";
     J
```

```
420  POKE 216,0:J = STRT
430  NEXT J
440  POKE 216,0
450  PRINT D$;"CLOSE";FLNME$
460  GOTO 100
470  REM   INITIALIZE REFERENCE F
     ILE:HOME
480  PRINT "INSERT REFS DISK INTO
      DRIVE 2,THEN PRESS <RETURN>
     "
490  GET CH$: PRINT R$
500 ST$ = VL$ + BNO$
510  PRINT D$;"OPEN REFS,L255,D2"

520  PRINT D$;"DELETE REFS"
530  PRINT D$;"OPEN REFS,L255"
540  PRINT D$;"WRITE REFS,R0"
550  PRINT ST$:ST$ = ""
560  PRINT D$;"CLOSE REFS"
570  END

]
```

```
10  REM  ***LSTFILE***
20 D$ =  CHR$ (4):R$ =  CHR$ (13)
     :T$ =  CHR$ (1)
30  HOME
40  PRINT "ENTER FILE NAME:  "
50  INPUT FLNAME$
60  PRINT "ENTER REC LENGTH:  ": INPUT
     RLN
70  INPUT "ENTER DRIVE NO:";DV$
80  PRINT "ENTER STRT REC NO:  ":
      INPUT STRT
90  PRINT "ENTER END REC NO:  ": INPUT
     ND
100  HOME
110  PRINT "INSERT DISK INTO DRIV
     E ";DV$;", THEN PRESS <RETUR
     N>"
120  GET CH$: PRINT R$
130  PRINT D$;"OPEN ";FLNAME$;",L
     ";RLN;",D";DV$
140  FOR I = STRT TO ND
150  ONERR  GOTO 210
160  PRINT D$;"READ";FLNAME$;",R"
     ;I
170  GOSUB 250: REM  * SUB GETCHA
     R *
180  FOR J = 1 TO 100: NEXT J
190  NEXT I
200  GOTO 230
210  REM    PEEK(222) CONTAINS ON
     ERR GOTO CODE FOR INTERRUPT
220  POKE 216,0
230  PRINT D$;"CLOSE";FLNAME$
240  END
250  REM
260  REM    * SUBRT GETCHAR *
270  REM
280  FOR K = 1 TO RLN
290  GET A$
300  IF A$ = R$ THEN  GOTO 330
310 ST$ = ST$ + A$
320  NEXT K
330  PRINT R$;I;":"
340  PRINT ST$
350 ST$ = ""
360  RETURN

]
```

```
 LOAD APP-CHNG-FILE
]LIST

10  REM     * APP-CHNG-FILE *
20  HOME
30 D$ =  CHR$ (4)
40 R$ =  CHR$ (13)
50 T$ =  CHR$ (1)
60 B$ =  CHR$ (7)
70 BLNK$ = "":TXT$ = BLNK$
80  REM  *USER INPUTS FILE PARAME
     TERS*
90  PRINT "ENTER FILENAME:  ": INPUT
     FLNAME$
100  PRINT "ENTER REC LENGTH*  ":
      INPUT RLN
110  INPUT "ENTER DRIVE NO:";DN%
120  PRINT "YOU HAVE INPUT ";FLNA
     ME$;" AND ";RLN
130  PRINT "OK? (Y/N)"
140  INPUT C$
150  IF C$ = "Y" THEN  GOTO 170
160  GOTO 90
170  PRINT D$;"OPEN ";FLNAME$;" ,
     L";RLN;",D";DN%
180  PRINT : PRINT "ENTER REC NO
     OR A -1 IF DONE"
190  INPUT RNUM
200  IF RNUM < 0 THEN  GOTO 400
210  PRINT "ENTER TEXT-END WITH A
      '*'"
220 J = 1
230  GET CH$
240  PRINT CH$;
250  IF CH$ <  > "*" THEN  GOTO 2
     90
260  IF TXT$ = BLNK$ THEN  GOTO 4
     00
270 JFLG = 1
280  GOTO 350
290 TXT$ = TXT$ + CH$
300 J = J + 1
310  IF J < RLN THEN  GOTO 230
320  PRINT B$
330  PRINT T$;TXT$: PRINT : PRINT

340  FOR IP = 1 TO 2000: NEXT IP
350  PRINT D$;"WRITE ";FLNAME$;",
     R";RNUM
360  PRINT TXT$
370  PRINT D$
380 TXT$ = BLNK$
390  GOTO 180
400  PRINT D$;"CLOSE";FLNAME$
410  PRINT TXT$
420  END

]
```

```
]LIST1,29

10  REM  BIBLIO
20  GOTO 7400: REM  MAIN PRGM


]
```

```
 LIST30,80

30  REM  SUBRT SETLN
40 LIN$ = ""
50 LG% = XL% -  LEN (XP$): IF LG%
     = 0 THEN 70
60  FOR YY = 1 TO LG%:LIN$ = LIN$
     + "0": NEXT YY
70 ST$ = ST$ + LIN$ + XP$
80  RETURN


]
```

```
 LIST90,1960

90  REM  SUBRT COMPHSHNO
100  DEF  FN MOD(A) =  INT ((A /
     C -  INT (A / C)) * C + .05)
      *  SGN (A / C)
110  REM  KW CK
120  IF FLNME$ = "KWTBL" THEN 240

130 AAFLG% = 0
140  FOR LP = 1 TO (ACTR% - 1)
150  IF AUTBL$(LP) = "" THEN 210
160  IF KE$ <  > AUTBL$(LP) THEN
     210
170  FOR Q = 1 TO 3
180 AK%(Q) = APROD%(LP,Q)
190  NEXT Q
200 AAFLG% = 1:LP = ACTR% - 1
210  NEXT LP
220  IF AAFLG% = 1 THEN 1480
230  GOTO 340
240 KFLG% = 0
250  FOR LP = 1 TO (KCTR% - 1)
260  IF KTBL$(LP) = "" THEN 320
270  IF KE$ <  > KTBL$(LP) THEN 3
     20
280  FOR Q = 1 TO 3
290 AK%(Q) = PROD%(LP,Q)
300  NEXT Q
310 KFLG% = 1:LP = KCTR% - 1
320  NEXT LP
330  IF KFLG% = 1 THEN 1480
340 LN% =  LEN (KE$)
350  FOR I = 1 TO LN%
360 CH$ =  MID$ (KE$,I,1)
370  FOR J = 1 TO 37
380  IF CH$ <  > KW$(1,J) THEN 41
     0
390 POLY$(I) = KW$(2,J)
400 J = 37
410  NEXT J
420  NEXT I
430  REM   CALCA
440 C = 2:AA$ = "":W = 2 ^ 16
450 A = ( SQR (5) - 1) * W / 2
460 A$ =  STR$ (A)
470 LG% =  LEN (A$)
480  FOR I = 1 TO LG%
490 CH$ =  MID$ (A$,I,1)
500  IF CH$ <  > "." THEN 520
510 I = LG%: GOTO 530
520 AA$ = AA$ + CH$
530  NEXT I
540 AA =  VAL (AA$)
550  REM   CONVRT TO BASE 2
560 I = 0
570 I = I + 1
580 QUOT% =  INT (AA / 2)
590 RM%(I) =  FN MOD(AA)
600  IF QUOT% = 0 THEN  GOTO 630
610 AA = QUOT%
620  GOTO 570
```

```
650  FOR J = 1 TO I
660 BIT%(J) = RM%(K)
670 K = K - 1
680  NEXT J
690 L2% = I
700  REM  BITMULT
710 HOLD% = 0:TEMP% = 0:ITME% = 0

720 BASE% = 64:L1% = LN%
730  FOR I = 1 TO L1%
740 R1%(I) = 0
750 SUM%(I) = 0
760 R3%(I) =  VAL (POLY$(I))
770  NEXT I
780  FOR I = 1 TO L2%
790 R2%(I) = BIT%(I)
800  NEXT I
810  REM  RPT ADDS
820  FOR ITME = 1 TO L2%
830  IF R2%(L2%) = 0 THEN 1090
840  FOR I = L1% TO 1 STEP  - 1
850 TEMP% = R1%(I) + R3%(I) + SUM
    %(I)
860  IF TEMP% >  = BASE% THEN 880

870 SUM%(I) = TEMP%: GOTO 940
880  REM  CARRY
890 SUM%(I) = TEMP% - BASE%
900  IF I <  > 1 THEN  GOTO 930
910 HOLD% = 1
920  GOTO 940
930 SUM%(I - 1) = 1
940  NEXT I
950  IF HOLD% > 0 THEN  GOTO 1000

960  FOR I = 1 TO L1%
970 R1%(I) = SUM%(I)
980  NEXT I
990  GOTO 1090
1000 R1%(1) = HOLD%
1010  FOR J = 2 TO (L1% + 1)
1020 R1%(J) = SUM%(J - 1)
1030  NEXT J
1040 L1% = L1% + 1
1050  FOR J = L1% TO 2 STEP  - 1
1060 R3%(J) = R3%(J - 1)
1070  NEXT J
1080 R3%(1) = 0
1090  REM  SHIFT
1100 NU%(1) = R1%(L1%)
1110 NX% = L2% - 1
1120  FOR J = 1 TO NX%
1130 NU%(J + 1) = R2%(J)
1140  NEXT J
1150  FOR J = 1 TO L2%
1160 R2%(J) = NU%(J)
1170 NU%(J) = 0
1180  NEXT J
1190 NX% = L1% - 1
1200  FOR J = 1 TO NX%
1210 NU%(J + 1) = R1%(J)
1220  NEXT J
1230 NU%(1) = 0
1240  FOR J = 1 TO L1%
```

```
1270 NU%(J) = 0
1280  NEXT J
1290 HOLD% = 0:TEMP% = 0
1300  NEXT ITME
1310 JJ = 1
1320  FOR J = (L2% - 2) TO L2%
1330 AK%(JJ) = R2%(J)
1340 JJ = JJ + 1
1350  NEXT J
1360  IF FLNME$ = "KWTBL" THEN 14
     30
1370 AUTBL$(ACTR%) = KE$
1380  FOR J = 1 TO (JJ - 1)
1390 APROD%(ACTR%,J) = AK%(J)
1400  NEXT J
1410 ACTR% = ACTR% + 1
1420  GOTO 1480
1430 KTBL$(KCTR%) = KE$
1440  FOR J = 1 TO (JJ - 1)
1450 PROD%(KCTR%,J) = AK%(J)
1460  NEXT J
1470 KCTR% = KCTR% + 1
1480  REM  CALC HSHNO
1490 SM% = 0:I% = 0:B% = 6:LN% =
     3
1500  REM   AK MOD W; W=2^16
1510  FOR J = LN% TO 1 STEP  - 1
1520 I% = I% + 1
1530  IF (SM% + B%) > 16 THEN 158
     0
1540 A1%(I%) = AK%(J)
1550 B1%(I%) = B%
1560 SM% = SM% + B%
1570  GOTO 1640
1580  REM  TAKE BITS
1590 MR% = 16 - SM%
1600 C = 2 ^ MR%
1610 A1%(I%) =  FN MOD(AK%(J))
1620 B1%(I%) = MR%
1630 J = 1
1640  NEXT J
1650 M% =  LOG (N%) /  LOG (2)
1660  REM   TAKE FRST MBITS
1670 SM% = 0:K% = 0
1680  FOR J = I% TO 1 STEP  - 1
1690 K% = K% + 1
1700  IF (SM% + B1%(J)) > M% THEN
     1760
1710 A2%(K%) = A1%(J)
1720 B2%(K%) = B1%(J)
1730  IF B1%(J) = M% THEN 1820
1740 SM% = SM% + B1%(J)
1750  GOTO 1830
1760  IF K% > 1 THEN  GOTO 1790
1770 MR% = B1%(J) - M%
1780  GOTO 1800
1790 MR% = B1%(J) - SM%
1800 A2%(K%) = A1%(J) / (2 ^ MR%)

1810 B2%(K%) = B1%(J) - MR%
1820 J = 1
1830  NEXT J
1840  REM  CALC HSHNO
1850 SM% = 0
```

```
1880 MLT% = 1
1890  IF I = KK THEN  GOTO 1930
1900  FOR J = (I + 1) TO K%
1910 MLT% = MLT% * (2 ^ B2%(J))
1920  NEXT J
1930 SM% = SM% + A2%(I) * MLT%
1940  NEXT I
1950 ADR% = SM%
1960  RETURN


]
```

```
 LIST1970,2110

1970  REM  SUBRT DSKIN
1980  PRINT D$;"CLOSE";FLNME$
1990  HOME : VTAB (5)
2000  PRINT "VOLUME MISMATCH-INSE
     RT DISK";DN%;"INTO DRIVE 1,
     THEN PRESS <RETURN>"
2010  GET CH$
2020  IF CH$ = R$ THEN  GOTO 2030
     : GOTO 2010
2030  PRINT R$: PRINT D$;"OPEN";F
     LNME$;",L";RLN
2040  PRINT D$;"READ";FLNME$;",R0
     "
2050  INPUT ST$
2060  IF DN% =  VAL ( LEFT$ (ST$,
     4)) THEN  GOTO 2080
2070  GOTO 2720
2080  ON AFLG% GOTO 2090,2100
2090 DV$(1) = "A" +  STR$ (DN%): GOTO
     2110
2100 DV$(1) = "K" +  STR$ (DN%)
2110  RETURN


]
```

```
LIST2120,2320

2120  REM  SUBRT CKTBL
2130  PRINT D$;"OPEN";FLNME$;",L"
     ;RLN
2140  REM  SCAN RECS
2150 STRT = ((ADR% * BS%) + 1) -
     ((DK% - 1) * (BPD% * BS%))
2160 ND = STRT + (BS% - 1)
2170 IREC =  - 1:FLG% = 0
2180  FOR I = STRT TO ND
2190  PRINT D$;"READ";FLNME$;",R"
     ;I
2200  INPUT ST$
2210 KY$ =  LEFT$ (ST$,20)
2220  IF KY$ <  > ZRO$ THEN 2270
2230  IF I < ND THEN  GOTO 2250
2240  IF HME% <  > ADR% THEN 2290

2250 IREC = I:I = ND
2260  GOTO 2290
2270  IF I < ND THEN  GOTO 2290
2280  IF LST% = 0 THEN LST% = I: REM
      PTR UPDT
2290  NEXT I
2300  IF IREC >  = 0 THEN 2320
2310 FLG% = 1
2320  RETURN


]
```

```
 LIST2330,2370

2330   REM   SUBRT LINPROB
2340 ADR% = ADR% - 1
2350   IF ADR% >   = LOL% THEN 2370

2360 ADR% = ADR% + LOL% + NB%
2370   RETURN


]
```

```
LIST2380,2450

2380  REM  SUBRT STOREHT
2390 OOVR$ = "0000"
2400  PRINT D$;"OPEN";FLNME$;",L"
     ;RLN
2410 ST$ = KE$ +  RIGHT$ (MTY$,(2
     0 -  LEN (KE$))) + VL$ + BCK
     $ + OOVR$
2420  PRINT D$;"WRITE";FLNME$;",R
     ";IREC
2430  PRINT ST$: PRINT
2440  PRINT D$;"CLOSE";FLNME$
2450  RETURN


]
```

```
LIST2460,2720

2460  REM  SUBRT UPDTE
2470  PRINT D$;"OPEN";FLNME$;",L"
     ;RLN
2480 ST$ = "":XL% = 2:XP$ =  STR$
     (DK%)
2490  GOSUB 30
2500 XP$ =  STR$ (IREC)
2510  GOSUB 30
2520 PTR$ = ST$
2530 UREC% = LST%
2540  PRINT D$;"READ";FLNME$;",R"
     ;UREC%
2550  INPUT ST$
2560 GD$ =  LEFT$ (ST$,25)
2570 OVR$ =  RIGHT$ (ST$,4)
2580 ST$ = ""
2590  IF OVR$ <  > "0000" THEN 26
     70
2600  REM  UPDT LST BLCKREC
2610 ST$ = GD$ + PTR$
2620  PRINT D$;"WRITE";FLNME$;",R
     ";UREC%
2630  PRINT ST$:ST$ = NLL$
2640 LST% = 0:PTR$ = NLL$
2650  PRINT D$;"CLOSE";FLNME$
2660  RETURN
2670  REM   FOLLOW PTR CHAIN
2680 DN% =  VAL ( LEFT$ (OVR$,2))

2690 UREC% =  VAL ( RIGHT$ (OVR$,
     2))
2700  IF DN% = DK% THEN  GOTO 254
     0
2710  GOSUB 1970
2720  GOTO 2540


]
```

```
LIST2730,2830

2730  REM  SUBRT PRFILE
2740 STRT = 1
2750 ND = (NUBLCK% * BS%) + BS%
2760  PRINT D$;"OPEN";FLNME$;",L"
     ;RLN
2770  FOR I = STRT TO ND
2780  PRINT D$;"READ";FLNME$;",R"
     ;I
2790  INPUT A$
2800  PRINT I: PRINT A$
2810  NEXT I
2820  PRINT D$;"CLOSE";FLNME$
2830  RETURN


]
```

```
LIST2840,2990

2840  REM  SUBRT HELP
2850  HOME : PRINT "INSERT SYSTEM
      MASTER INTO DRV 1,THEN PRES
     S <RETURN>"
2860  GET CH$
2870  IF CH$ = R$ THEN 2890
2880  GOTO 2860
2890  PRINT R$:RLN = 254: PRINT D
     $;"OPEN DESCRIP,L254,D1"
2900  FOR JL = JB TO JE
2910  PRINT D$;"READ DESCRIP,R";J
     L
2920  GOSUB 6050: PRINT ST$:ST$ =
     ""
2930  NEXT JL:ST$ = ""
2940  PRINT D$;"CLOSE DESCRIP"
2950  PRINT "INSERT PREVIOUS DISK
      INTO DRV 1, THEN PRESS <RET
     URN>"
2960  GET CH$
2970  IF CH$ = R$ THEN 2990
2980  GOTO 2960
2990  PRINT T$: RETURN


]
```

```
 LIST3000,3040

3000   REM   SUBRT PUSH
3010 TTP% = TTP% + 1
3020 S2$(TTP%) = KY$
3030 KY$ = ""
3040   RETURN


]
```

```
 LIST3050,3120

3050  REM  SUBRT GETKW
3060 ST$ = ""
3070  FOR LP = 20 TO 1 STEP  - 1
3080  IF  MID$ (KY$,LP,1) = " " THEN
      GOTO 3100
3090 FIN = LP:LP = 1
3100  NEXT LP
3110 ST$ =  LEFT$ (KY$,FIN)
3120  RETURN


]
```

```
 LIST3130,3420

3130  REM  SUBRT GETLSTREC
3140 UREC% = LST%
3150  PRINT D$;"READ";FLNME$;",R"
     ;UREC%
3160  INPUT ST$
3170 NXT% =  VAL ( RIGHT$ (ST$,4)
     )
3180  IF NXT% = 0 THEN  GOTO 3210

3190 UREC% = NXT%
3200  GOTO 3150
3210 LAST% = UREC%
3220  RETURN
3230  REM   SUBRT VRFYDSK
3240  IF DK% =  VAL ( MID$ (DV$(1
     ),2)) THEN  GOTO 3420
3250  PRINT D$;"CLOSE";FLNME$
3260  HOME : VTAB 10
3270  ON AFLG% GOTO 3280,3300
3280  PRINT "VOLUME MISMATCH-INSE
     RT DISK A";DK%;" INTO DRIVE
     1, THEN PRESS <RETURN>"
3290  GOTO 3310
3300  PRINT "VOLUME MISMATCH-INSE
     RT DISK K";DK%;" INTO DRIVE
     1, THEN PRESS <RETURN>"
3310  GET CH$
3320  IF CH$ = R$ THEN  GOTO 3340

3330  GOTO 3310
3340  PRINT T$: PRINT D$;"OPEN";F
     LNME$;",L";RLN
3350  PRINT D$;"READ";FLNME$;",RO
     "
3360  INPUT ST$
3370  IF DK% =  VAL ( LEFT$ (ST$,
     4)) THEN  GOTO 3390
3380  GOTO 3270
3390  ON AFLG% GOTO 3400,3410
3400 DV$(1) = "A" +  STR$ (DK%): GOTO
     3420
3410 DV$(1) = "K" +  STR$ (DK%)
3420  RETURN


]
```

```
LIST3230,3420

3230  REM   SUBRT VRFYDSK
3240  IF DK% =  VAL ( MID$ (DV$(1
     ),2)) THEN  GOTO 3420
3250  PRINT D$;"CLOSE";FLNME$
3260  HOME : VTAB 10
3270  ON AFLG% GOTO 3280,3300
3280  PRINT "VOLUME MISMATCH-INSE
     RT DISK A";DK%;" INTO DRIVE
     1, THEN PRESS <RETURN>"
3290  GOTO 3310
3300  PRINT "VOLUME MISMATCH-INSE
     RT DISK K";DK%;" INTO DRIVE
     1, THEN PRESS <RETURN>"
3310  GET CH$
3320  IF CH$ = R$ THEN  GOTO 3340

3330  GOTO 3310
3340  PRINT T$: PRINT D$;"OPEN";F
     LNME$;",L";RLN
3350  PRINT D$;"READ";FLNME$;",R0
     "
3360  INPUT ST$
3370  IF DK% =  VAL ( LEFT$ (ST$,
     4)) THEN  GOTO 3390
3380  GOTO 3270
3390  ON AFLG% GOTO 3400,3410
3400 DV$(1) = "A" +  STR$ (DK%): GOTO
     3420
3410 DV$(1) = "K" +  STR$ (DK%)
3420  RETURN


]
```

```
]
 LIST3430,6040

3430  REM  SUBRT LVHI
3440 X =  FRE (0)
3450  PRINT D$;"OPEN";FLNME$;",L"
     ;RLN
3460  ON AFLG% GOTO 3470,3500
3470 XREC% = IA%
3480  FOR K = 1 TO XREC%:CCE$(K) =
     NME$(K): NEXT K
3490  GOTO 3520
3500 XREC% = IK%
3510  FOR K = 1 TO XREC%:CCE$(K) =
     CE$(K): NEXT K
3520  FOR WREC = 1 TO XREC%
3530 KE$ = CCE$(WREC)
3540  PRINT WREC;":";KE$
3550 NREC% = NREC% + 1
3560 H%(JPTR%) = NB%
3570 N% = H%(JPTR%)
3580  GOSUB 90
3590 H2% = ADR%
3600  IF JPTR% > 0 THEN 3620
3610 TMP% =  - 1: GOTO 3660
3620 N% = H%(JPTR% - 1)
3630  GOSUB 90
3640 H1% = ADR%
3650 TMP% = H1%
3660  IF P% = 0 THEN 3690
3670  IF TMP% < P% THEN 3690
3680 ADR% = H1%: GOTO 3700
3690 ADR% = H2%
3700  PRINT "ADR=";ADR%
3710 HME% = ADR%
3720 DK% = ( INT (ADR% / BPD%)) +
     1
3730  GOSUB 3230
3740 LOL% = (DK% - 1) * BPD%:UPL%
      = LOL% + (BPD% - 1)
3750  ON AFLG% GOTO 3760,3770
3760 DV$(1) = "A" +  STR$ (DK%): GOTO
     3780
3770 DV$(1) = "K" +  STR$ (DK%)
3780  GOSUB 2120
3790  IF FLG% > 0 THEN 3830
3800 VL$ = VOL$
3810  GOSUB 2380
3820  GOTO 5970
3830 ALPHA = NREC% / (NB% * BS%)
3840  IF ALPHA > TALPHA GOTO 3920

3850  REM   STORE R AS OVRFLW FRO
     M ADR
3860  GOSUB 2330: GOSUB 2120
3870  IF FLG% = 0 THEN 3890
3880  GOTO 3860
3890 VL$ = VOL$: GOSUB 2380
3900  GOSUB 2460
3910  GOTO 5970
3920  REM   ALLOC NEW BLOCK
3930 TMP% = H%(JPTR%)
3940  IF P% = 0 THEN JPTR% = JPTR
```

```
     1) * V%)
3960 NB% = NB% + 1
3970 H%(JPTR%) = NB%
3980 H%(JPTR% - 1) = TMP%
3990 ADR% = NUBLCK%
4000  IF NUBLCK% > UPL% THEN 4020

4010  GOTO 4310
4020  ON AFLG% GOTO 4030,4040
4030 DTA%(1) = DTA%(1) + 1:DK% =
     DTA%(1): GOTO 4050
4040 DTA%(7) = DTA%(7) + 1:DK% =
     DTA%(7)
4050  PRINT D$;"CLOSE";FLNME$
4060  HOME : VTAB (5)
4070  PRINT "DISK FULL-INSERT NEW
      DISK IN DRIVE 1, THEN PRESS
      <RETURN>"
4080  GET CH$
4090  IF CH$ = R$ THEN 4110
4100  GOTO 4080
4110  PRINT R$: PRINT D$;"OPEN";F
     LNME$;",L";RLN
4120 ST$ = "":XL% = 4
4130  ON AFLG% GOTO 4140,4150
4140 XP$ =  STR$ (DTA%(1)): GOTO
     4160
4150 XP$ =  STR$ (DTA%(7))
4160  GOSUB 30
4170  PRINT D$;"WRITE";FLNME$;",R
     0"
4180  PRINT ST$:ST$ = ""
4190 ST$ = ZRO$ + "000000000"
4200  FOR QK = 1 TO (BPD% * BS%)
4210  PRINT D$;"WRITE";FLNME$;",R
     ";QK
4220  PRINT ST$
4230  NEXT QK
4240  ON AFLG% GOTO 4250,4260
4250 DV$(1) = "A" +  STR$ (DTA%(1
     )): GOTO 4270
4260 DV$(1) = "K" +  STR$ (DTA%(7
     ))
4270  PRINT D$;"CLOSE";FLNME$
4280  GOTO 4310
4290  REM   STORE R AS OVRFLW
4300  GOSUB 2330
4310  GOSUB 2120
4320  IF FLG% = 0 THEN 4340
4330  GOTO 4300
4340 VL$ = VOL$: GOSUB 2380: GOSUB
     2460
4350  REM   FOREIGNER CHK
4360 DK% = ( INT (P% / BPD%) + 1)

4370  GOSUB 3230
4380 N% = H%(JPTR% - 1)
4390 C% = 0:CT% = 0
4400 STRT% = ((P% * BS%) + 1) - (
     (DK% - 1) * (BPD% * BS%))
4410 ND% = STRT% + (BS% - 1)
4420 NXTREC = 0
4430  PRINT D$;"OPEN";FLNME$;",L"
     ;RLN
```

```
    ;JREC
4460  INPUT ST$
4470 KY$ =  LEFT$ (ST$,25)
4480 OVR$ =  RIGHT$ (ST$,4)
4490  IF ( LEFT$ (KY$,20)) <  > Z
    RO$ THEN 4510
4500  GOTO 4670
4510 KKY$ = KY$:KY$ =  LEFT$ (KKY
    $,20)
4520  GOSUB 3050
4530 KE$ = ST$
4540  GOSUB 90
4550  IF ADR% = P% THEN 4590
4560  REM   SAVE FOREIGNERS
4570 C% = C% + 1:SVE$(C%,2) = KKY
    $ + "0000":SVE$(C%,3) =  STR$
    (ADR%):SVE$(C%,1) =  STR$ (J
    REC)
4580  GOTO 4610
4590 KY$ = KKY$
4600  GOSUB 3000
4610  IF JREC < ND% THEN 4670
4620  IF OVR$ = "0000" THEN 4670
4630 DN% =  VAL ( LEFT$ (OVR$,2))

4640 NXTREC =  VAL ( RIGHT$ (OVR$
    ,2))
4650 TP% = TP% + 1
4660 S1%(TP%,1) = DN%:S1%(TP%,2) =
    NXTREC
4670  NEXT JREC
4680  IF NXTREC = 0 THEN 4960
4690  REM   GET OVRFLW KW
4700  IF DK% = DN% THEN 4720
4710  GOSUB 1970:DK% = DN%
4720  PRINT D$;"READ";FLNME$;",R"
    ;NXTREC
4730  INPUT ST$
4740 KY$ =  LEFT$ (ST$,25)
4750 OVR$ =  RIGHT$ (ST$,4)
4760  IF ( LEFT$ (KY$,20)) = ZRO$
     THEN  GOTO 4900
4770 KKY$ = KY$:KY$ =  LEFT$ (KKY
    $,20)
4780  GOSUB 3050
4790 KE$ = ST$
4800  GOSUB 90
4810  IF ADR% <  > P% THEN 4840
4820 KY$ = KKY$
4830  GOTO 4890
4840 C% = C% + 1
4850 SVE$(C%,1) =  STR$ (NXTREC)
4860 SVE$(C%,2) = KKY$ + "0000"
4870 SVE$(C%,3) =  STR$ (ADR%)
4880  GOTO 4900
4890  GOSUB 3000
4900 DN% =  VAL ( LEFT$ (OVR$,2))

4910 NXTREC =  VAL ( RIGHT$ (OVR$
    ,2))
4920  IF NXTREC = 0 THEN 4960
4930 TP% = TP% + 1
4940 S1%(TP%,1) = DN%:S1%(TP%,2) =
    NXTREC
```

```
4970  REM   REVISE CHAINS FROM BL
     CK P
4980  PRINT D$;"OPEN";FLNME$;",L"
     ;RLN
4990  FOR U = 1 TO C%
5000 SADR% =  VAL (SVE$(U,3))
5010 DK% = ( INT (SADR% / BPD%) +
     1)
5020  GOSUB 3230
5030 FIN% = ((SADR% * BS%) + 1) +
     (BS% - 1)
5040 ST$ = NLL$
5050  PRINT D$;"READ";FLNME$;",R"
     ;FIN%
5060  INPUT ST$
5070 GD$ =  LEFT$ (ST$,25):OVR$ =
      RIGHT$ (ST$,4)
5080  IF OVR$ = "0000" THEN 5220
5090 NXT% =  VAL ( RIGHT$ (OVR$,2
     ))
5100  FOR V = 1 TO C%
5110  IF NXT% <  >  VAL (SVE$(V,1
     )) THEN 5200
5120  PRINT D$;"READ";FLNME$;",R"
     ;NXT%
5130  INPUT ST$
5140 OVR$ =  RIGHT$ (ST$,4)
5150 ST$ = NLL$
5160 ST$ = GD$ + OVR$: PRINT ST$
5170  PRINT D$;"WRITE";FLNME$;",R
     ";FIN%
5180  PRINT ST$:ST$ = NLL$
5190 V = C%
5200  NEXT V
5210 FIN% = NXT%: GOTO 5040
5220  NEXT U
5230  PRINT D$;"CLOSE";FLNME$
5240  REM  INIT P
5250 SX$ = ZRO$ + "000000000"
5260  PRINT D$;"OPEN";FLNME$;",L"
     ;RLN
5270 DK% = ( INT (P% / BPD%) + 1)

5280  GOSUB 3230
5290  FOR LREC = STRT% TO ND%
5300  PRINT D$;"WRITE";FLNME$;",R
     ";LREC
5310  PRINT SX$
5320  NEXT LREC
5330  IF TP% = 0 THEN 5420
5340  FOR L = 1 TO TP%
5350 DN% = S1%(L,1)
5360  IF DK% = DN% THEN 5380
5370  GOSUB 1970:DK% = DN%
5380 LREC = S1%(L,2)
5390  PRINT D$;"WRITE";FLNME$;",R
     ";LREC
5400  PRINT SX$
5410  NEXT L
5420  PRINT D$;"CLOSE";FLNME$
5430  IF TTP% = 0 THEN 5740
5440  REM   REHSH P&OVRFLWS
5450 N% = H%(JPTR%)
5460  REM  POP STK2
```

```
5480 KY$ =  LEFT$ (S2$(L),20)
5490 VL$ =  MID$ (S2$(L),21,2)
5500 BCK$ =  MID$ (S2$(L),23,3)
5510  GOSUB 3050
5520 KE$ = ST$:ST$ = ""
5530  GOSUB 90
5540 LST% = 0
5550 HME% = ADR%
5560  GOSUB 3230
5570  GOSUB 2120
5580  IF FLG% = 0 THEN 5610
5590  GOSUB 2330
5600  GOTO 5570
5610  GOSUB 2380
5620  IF LST% = 0 THEN 5640
5630  GOSUB 2460
5640  NEXT L
5650  FOR L = 1 TO TTP%
5660 S2$(L) = ""
5670  NEXT L
5680 TTP% = 0
5690  PRINT D$;"CLOSE";FLNME$
5700  FOR L = 1 TO TP%
5710 S1%(L,1) = 0:S1%(L,2) = 0
5720  NEXT L
5730 TP% = 0
5740 LST% = 0
5750  IF C% = 0 THEN 5950
5760  REM   DEAL WITH FOREIGNER R
     ECORDS
5770  PRINT D$;"OPEN";FLNME$;",L"
     ;RLN
5780  FOR IC = 1 TO C%
5790 ADR% =  VAL (SVE$(IC,3))
5800 LST% = 0
5810 HME% = ADR%
5820 DK% = ( INT (ADR% / BPD%) +
     1)
5830  GOSUB 3230
5840  GOSUB 2120
5850  IF FLG% = 0 THEN 5880
5860  GOSUB 2330
5870  GOTO 5840
5880  PRINT D$;"WRITE";FLNME$;",R
     ";IREC
5890  PRINT SVE$(IC,2)
5900  PRINT D$;"CLOSE";FLNME$
5910  IF LST% = 0 THEN 5930
5920  GOSUB 2460
5930  NEXT IC
5940  FOR KT = 1 TO C%:SVE$(KT,1)
      = "":SVE$(KT,2) = "":SVE$(K
     T,3) = "": NEXT KT
5950 P% = P% + 1
5960  IF P% >  = (2 ^ (JPTR% - 1)
      * V%) THEN P% = 0
5970 LST% = 0
5980  IF JPTR% = 0 THEN 6010
5990  PRINT "P,J:";P%,JPTR%
6000  PRINT "H(J),H(J-1):";H%(JPT
     R%),H%(JPTR% - 1): PRINT
6010  NEXT WREC
6020  PRINT D$;"CLOSE";FLNME$
6030  PRINT : GOSUB 2730: PRINT
```

```
 LIST6050,6120

6050  REM  SUBRT GETCHAR
6060 ST$ = ""
6070  FOR K = 1 TO RLN
6080  GET A$
6090  IF A$ = R$ THEN 6120
6100 ST$ = ST$ + A$
6110  NEXT K
6120  PRINT T$: RETURN


]
```

```
]LIST6130,7240

6130  REM  SUBRT KWSERCH
6140 RLN = 31
6150  IF IA% = 0 THEN  GOTO 6250
6160 AFLG% = 1:FLNME$ = "AWTBL"
6170 MDSKS% = DTA%(1)
6180 NREC% = DTA%(2)
6190 NB% = DTA%(3)
6200 P% = DTA%(4)
6210 JPTR% = DTA%(5)
6220  IF JPTR% = 0 THEN 6340
6230 H%(JPTR% - 1) = DTA%(6)
6240  GOTO 6340
6250  IF IK% = 0 THEN 7240
6260 AFLG% = 2:FLNME$ = "KWTBL"
6270 MDSKS% = DTA%(7)
6280 NREC% = DTA%(8)
6290 NB% = DTA%(9)
6300 P% = DTA%(10)
6310 JPTR% = DTA%(11)
6320  IF JPTR% = 0 THEN 6340
6330 H%(JPTR% - 1) = DTA%(12)
6340  ON AFLG% GOTO 6350,6400
6350 XREC% = IA%
6360  FOR K = 1 TO XREC%
6370 CCE$(K) = NME$(K)
6380  NEXT K
6390  GOTO 6440
6400 XREC% = IK%
6410  FOR K = 1 TO XREC%
6420 CCE$(K) = CE$(K)
6430  NEXT K
6440 BFR% = 0
6450  FOR WREC = 1 TO XREC%
6460 KE$ = CCE$(WREC)
6470  PRINT WREC;":";KE$
6480 H%(JPTR%) = NB%
6490 N% = H%(JPTR%)
6500  GOSUB 90
6510 DK% = ( INT (ADR% / BPD%)) +
     1
6520  HOME : VTAB 5
6530  ON AFLG% GOTO 6540,6560
6540  IF BFR% = DK% THEN 6610
6550  PRINT "INSERT AWTBL DISK ";
     DK%;" INTO DRIVE 1, THEN PRE
     SS <RETURN>": GOTO 6580
6560  IF BFR% = DK% THEN 6610
6570  PRINT "INSERT KWTBL DISK ";
     DK%;"INTO DRIVE 1, THEN PRES
     S <RETURN>"
6580  GET CH$
6590  IF CH$ = R$ THEN 6610
6600  GOTO 6580
6610  PRINT T$: PRINT D$;"OPEN";F
     LNME$;",L";RLN;",D1"
6620  PRINT D$;"READ";FLNME$;",R0
     "
6630  INPUT ST$
6640  IF DK% =  VAL ( LEFT$ (ST$,
     4)) THEN 6670
```

```
6670  ON AFLG% GOTO 6680,6690
6680 DV$(1) = "A" +  STR$ (DK%): GOTO
     6700
6690 DV$(1) = "K" +  STR$ (DK%)
6700  REM  SRCH HSH TBL
6710 STRT% = ((ADR% * BS%) + 1) -
     ((DK% - 1) * (BPD% * BS%))
6720 ND% = STRT% + (BS% - 1)
6730 NXTREC = 0
6740  PRINT D$;"OPEN";FLNME$;",L"
     ;RLN;",D1"
6750  FOR IREC = STRT% TO ND%
6760  PRINT D$;"READ";FLNME$;",R"
     ;IREC
6770  INPUT ST$
6780 VOL$ =  MID$ (ST$,21,2)
6790 BLCK$ =  MID$ (ST$,23,3)
6800 OVR$ =  RIGHT$ (ST$,4)
6810 KY$ = ""
6820  FOR L = 20 TO 1 STEP  - 1
6830  IF  MID$ (ST$,L,1) = " " THEN
      GOTO 6850
6840 FIN = L:L = 1
6850  NEXT L
6860 KY$ =  LEFT$ (ST$,FIN)
6870  IF KE$ <  > KY$ THEN 6910
6880 X% = X% + 1
6890 ANS$(X%,1) =  STR$ (IREC)
6900 ANS$(X%,2) = KE$ + VOL$ + BL
     CK$
6910  IF IREC < ND% THEN 6950
6920  IF OVR$ = "0000" THEN 6950
6930 DN% =  VAL ( LEFT$ (OVR$,2))

6940 NXTREC =  VAL ( RIGHT$ (OVR$
     ,2))
6950  NEXT IREC
6960  IF NXTREC = 0 THEN 7190
6970  REM  OVRFLWS
6980  IF DK% = DN% THEN 7010
6990  GOSUB 1970
7000 DK% = DN%
7010  PRINT D$;"READ";FLNME$;",R"
     ;NXTREC
7020  INPUT ST$
7030 VOL$ =  MID$ (ST$,21,2)
7040 BLCK$ =  MID$ (ST$,23,3)
7050 OVR$ =  RIGHT$ (ST$,4)
7060 KY$ = ""
7070  FOR L = 20 TO 1 STEP  - 1
7080  IF  MID$ (ST$,L,1) = " " THEN
     7100
7090 FIN = L:L = 1
7100  NEXT L
7110 KY$ =  LEFT$ (ST$,FIN)
7120  IF KY$ <  > KE$ THEN 7150
7130 X% = X% + 1
7140 ANS$(X%,1) =  STR$ (NXTREC):
     ANS$(X%,2) = KE$ + VOL$ + BL
     CK$
7150  IF OVR$ = "0000" THEN 7190
7160 DN% =  VAL ( LEFT$ (OVR$,2))

7170 NXTREC =  VAL ( RIGHT$ (OVR$
```

```
7190   PRINT D$;"CLOSE";FLNME$
7200 BFR% = DK%
7210   NEXT WREC
7220   IF X% = 0 THEN 7240
7230 KY$ = ""
7240   RETURN

]
```

```
]LIST

7400  REM  BIBLIOGRAF
7410  CLEAR : HOME
7420  DATA    A,B,C,D,E,F,G,H,I,J
      ,K,L,M,N,O,P,Q,R,S,T,U,V,W,X
      ,Y,Z,0,1,2,3,4,5,6,7,8,9,.
7430  DATA    0,1,2,3,4,5,6,7,8,9
      ,10,11,12,13,14,15,16,17,18,
      19,20,21,22,23,24,25,26,27,2
      8,29,30,31,32,33,34,35,36
7440 D$ =  CHR$ (4):R$ =  CHR$ (1
      3):T$ =  CHR$ (1)
7450  DIM H%(30),S1%(10,2),S2$(20
      )
7460  DIM KTBL$(30),PROD%(30,3),A
      UTBL$(30),APROD%(30,3)
7470  DIM SVE$(10,3)
7480  DIM KW$(2,40),POLY$(21)
7490  DIM RM%(21),BIT%(21)
7500  DIM R1%(21),R2%(21),R3%(21)
      ,SUM%(21),NU%(21)
7510  DIM A1%(21),A2%(21),B1%(21)
      ,B2%(21),AK%(4)
7520  DIM ANS$(10,2),NME$(10),CE$
      (10),KP$(10),CCE$(10)
7530  DIM DV$(2),DTA%(15)
7540  DIM RF$(10)
7550 BS% = 3:TALPHA = 0.8:TP% = 0
      :TTP% = 0:X% = 0:KCTR% = 1:A
      CTR% = 1:BPD% = 8:XND% = 24
7560 ZRO$ = "00000000000000000000
      ":MTY$ = "
        "
7570 NLL$ = "":REF$ = NLL$:BUFF$ =
      NLL$:SFLG% = 0:BFLG% = 0
7580  FOR I = 1 TO 2
7590  FOR J = 1 TO 37
7600  READ DD$
7610 KW$(I,J) = DD$
7620  NEXT J
7630  NEXT I
7640 KW$(1,38) = " "
7650 KW$(2,38) = "37"
7660  RESTORE : PRINT
7670  ONERR  GOTO 15160
7680  REM  LOAD PTRS
7690 STRT% = 1:ND% = 6:IREC = 0
7700  PRINT D$;"OPEN INDXPTRS,L31
      ,D1"
7710  PRINT D$;"READ INDXPTRS,R";
      IREC
7720  INPUT ST$
7730 MDSKS% =  VAL ( LEFT$ (ST$,4
      ))
7740 NREC% =  VAL ( MID$ (ST$,5,4
      ))
7750 NB% =  VAL ( MID$ (ST$,9,4))

7760 P% =  VAL ( MID$ (ST$,13,4))
```

```
7780  IF JPTR% = 0 THEN 7800
7790 H%(JPTR% - 1) =  VAL ( RIGHT$
     (ST$,4))
7800 V% = NB%
7810  FOR J = STRT% TO ND%:
7820  IF J <  = 6 THEN 7850
7830 JJ = J - 6
7840  ON JJ GOTO 7860,7870,7880,7
     890,7900,7910
7850  ON J GOTO 7860,7870,7880,78
     90,7900,7910
7860 DTA%(J) = MDSKS%: GOTO 7940
7870 DTA%(J) = NREC%: GOTO 7940
7880 DTA%(J) = NB%: GOTO 7940
7890 DTA%(J) = P%: GOTO 7940
7900 DTA%(J) = JPTR%: GOTO 7940
7910  IF JPTR% > 0 THEN 7930
7920 DTA%(J) = 0: GOTO 7940
7930 DTA%(J) = H%(JPTR% - 1)
7940  NEXT J
7950 ST$ = ""
7960 IREC = IREC + 1
7970  ON IREC GOTO 7980,8010
7980  REM  LOAD KWPTRS
7990 STRT% = 7:ND% = 12
8000  GOTO 7710
8010  REM  LOAD REFPTRS
8020  PRINT D$;"READ INDXPTRS,R";
     IREC
8030  INPUT ST$
8040 MDSKS% =  VAL ( LEFT$ (ST$,4
     ))
8050 VOL% =  VAL ( MID$ (ST$,5,2)
     )
8060 BNO% =  VAL ( MID$ (ST$,7,3)
     )
8070  FOR J = 13 TO 15
8080 JJ = J - 12
8090  ON JJ GOTO 8100,8110,8120
8100 DTA%(J) = MDSKS%: GOTO 8130
8110 DTA%(J) = VOL%: GOTO 8130
8120 DTA%(J) = BNO%
8130  NEXT J
8140  PRINT D$;"CLOSE INDXPTRS"
8150  PRINT D$;"OPEN DESCRIP,L254
     ,D1"
8160  IF BNO% = 1 THEN 8270
8170  HOME : VTAB 5
8180  PRINT "INSERT DISK R";DTA%(
     13);" INTO DRIVE 2, THEN PRE
     SS <RETURN>"
8190  GET CH$
8200  IF CH$ = R$ THEN 8220
8210  GOTO 8190
8220  PRINT R$:ST$ = ""
8230  PRINT D$;"OPEN REFS,L255,D2
     "
8240  PRINT D$;"READ REFS,R";DTA%
     (15)
8250  INPUT ST$:REF$ = ST$:ST$ =
     ""
8260  PRINT D$;"CLOSE REFS"
8270  HOME : VTAB 11
8280  PRINT  TAB( 10);"BIBLIOGRAP
```

```
8300  PRINT  TAB( 12);"RETRIEVAL
     SYSTEM"
8310  FOR K = 1 TO 1000: NEXT K
8320  HOME : REM  MENU1
8330  VTAB 5: PRINT "SYSTEM OPTIO
     NS:"
8340  VTAB 10: PRINT "SYSTEM DESC
     RIPTION"; SPC( 17);"D"
8350  VTAB 12
8360  PRINT "ENTER REFERENCE"; SPC(
     20);"E"
8370  VTAB 14
8380  PRINT "SEARCH REFERENCE"; SPC(
     19);"S"
8390  VTAB 16
8400  PRINT "QUIT"; SPC( 31);"Q"
8410  VTAB 22
8420  INPUT "ENTER KEY:";CY$
8430  IF CY$ = "D" THEN 8530
8440  IF CY$ = "E" THEN 8570
8450  IF CY$ = "S" THEN 13100
8460  IF CY$ = "Q" THEN 8480
8470  GOTO 8410
8480  HOME : VTAB 5:RSP$ = NLL$
8490  INPUT "VERIFY QUIT!(Y/N)";R
     SP$
8500  IF RSP$ = "Y" THEN 15160
8510  IF RSP$ = "N" THEN 8320
8520  GOTO 8410
8530  REM  SYSTM DSCRP
8540 JB = 1:JE = 5: GOSUB 2840
8550  FOR K = 1 TO 2000: NEXT K
8560  GOTO 8320
8570  REM  MENU2
8580  HOME : VTAB 5
8590  PRINT "INSERT DISK A";DTA%(
     1);" INTO DRIVE 1"
8600 DV$(1) = "A" +  STR$ (DTA%(1
     ))
8610  VTAB 8
8620  PRINT "INSERT DISK R";DTA%(
     13);" INTO DRIVE 2, THEN PRE
     SS <RETURN>"
8630 DV$(2) = "R" +  STR$ (DTA%(1
     3))
8640  GET CH$
8650  IF CH$ = R$ THEN  GOTO 8670

8660  GOTO 8640
8670  PRINT R$:TXT$ = "":JRNL$ =
     "":VVL$ = "":YR$ = "":PGS$ =
     ""
8680 RFLG% = 0:X =  FRE (0)
8690  HOME : VTAB 5: PRINT "ENTER
      REFERENCE:"
8700  VTAB 8
8710  PRINT "HELP"; SPC( 31);"H"
8720  VTAB 10
8730  PRINT "INPUT"; SPC( 30);"I"

8740  VTAB 12
8750  PRINT "CHECK DATA"; SPC( 25
     );"C"
8760  VTAB 14
```

```
8780  VTAB 16
8790  PRINT "QUIT"; SPC( 31);"Q"
8800  VTAB 22
8810  INPUT "ENTER KEY:";CY$
8820  IF CY$ = "H" THEN 8930
8830  IF CY$ = "I" THEN 8960
8840  IF CY$ = "C" THEN 9630
8850  IF CY$ = "S" THEN 11010
8860  IF CY$ = "Q" THEN 8880
8870  GOTO 8800
8880  HOME : VTAB 5:RSP$ = NLL$
8890  INPUT "VERIFY QUIT! (Y/N)";
     RSP$
8900  IF RSP$ = "Y" THEN 8560
8910  IF RSP$ = "N" THEN 8680
8920  GOTO 8880
8930 JB = 6:JE = 14: GOSUB 2840
8940  FOR K = 1 TO 2000: NEXT K
8950  GOTO 8690
8960  REM   INPUT REF
8970  IF RFLG% = 0 THEN 9040
8980  HOME : VTAB 5:RSP$ = ""
8990  PRINT "REFERENCE HAS ALREAD
     Y BEEN INPUT-DO YOU WANT TO
     REENTER IT? (Y/N)"
9000  INPUT RSP$
9010  IF RSP$ = "Y" THEN 9040
9020  IF RSP$ = "N" THEN 8690
9030  GOTO 8980
9040  HOME
9050  VTAB 2
9060  PRINT " AUTHORS:": HTAB 1
9070  VTAB 7
9080  PRINT " TITLE:": HTAB 1
9090  VTAB 15
9100  PRINT " JOURNAL:": HTAB 1
9110  VTAB 18
9120  PRINT " VOLUME:": HTAB 1
9130  VTAB 20
9140  PRINT " YEAR:": HTAB 1
9150  VTAB 22
9160  PRINT " PAGES:": HTAB 1
9170  REM   CHANGE TEXT WINDOW
9180  POKE 33,29
9190  POKE 32,11
9200  POKE 34,1
9210  POKE 35,6
9220  CALL  - 936
9230 IA% = 1
9240  INPUT NME$(IA%)
9250  IF NME$(IA%) = "" THEN  GOTO
     9270
9260 IA% = IA% + 1: GOTO 9240
9270 IA% = IA% - 1
9280  POKE 34,6
9290  POKE 35,14
9300  CALL  - 936
9310  INPUT TXT$
9320  POKE 34,14
9330  POKE 35,17
9340  CALL  - 936
9350  INPUT JRNL$
9360  POKE 34,17
9370  POKE 35,19
```

```
9400  POKE 34,19
9410  POKE 35,21
9420  CALL  - 936
9430  INPUT YR$
9440  POKE 34,21
9450  POKE 35,23
9460  CALL  - 936
9470  INPUT PGS$
9480  TEXT
9490  FOR K = 1 TO 300: NEXT K
9500  HOME : VTAB 2
9510  PRINT "KEYWORDS:": HTAB 1
9520  POKE 33,29
9530  POKE 32,11
9540  POKE 34,1
9550  POKE 35,23
9560  CALL  - 936
9570 IK% = 1
9580  INPUT CE$(IK%)
9590  IF CE$(IK%) = "" THEN  GOTO
     9610
9600 IK% = IK% + 1: GOTO 9580
9610 IK% = IK% - 1
9620  TEXT :RFLG% = 1
9630  REM  CHCKDATA:RSP$=""
9640  HOME : VTAB 5: PRINT "BEFOR
     E YOU STORE THIS DATA ON FIL
     E, YOU SHOULD CHECK IT FOR E
     NTRY ERRORS."
9650  VTAB 10: INPUT "CHECKDATA (
     Y/N)?";RSP$
9660  IF RSP$ = "Y" THEN 9680
9670  GOTO 8690
9680  HOME : PRINT "CHECK OF INPU
     T DATA:"; SPC( 14);"KEY"
9690  VTAB 2
9700  PRINT "AUTHORS:"; SPC( 27);
     "A"
9710  VTAB 7
9720  PRINT "TITLE:"; SPC( 29);"T
     "
9730  VTAB 12
9740  PRINT "JOURNAL:"; SPC( 27);
     "J"
9750  VTAB 14
9760  PRINT "VOLUME:"; SPC( 28);"
     V"
9770  VTAB 16
9780  PRINT "PAGES:"; SPC( 29);"P
     "
9790  VTAB 18
9800  PRINT "YEAR:"; SPC( 30);"R"

9810  VTAB 20
9820  PRINT "KEYWORDS:"; SPC( 26)
     ;"K"
9830  POKE 33,20
9840  POKE 32,10
9850  POKE 34,1
9860  POKE 35,6
9870  CALL  - 936
9880 STRT% = 1
9890 ND% = STRT% + 3
9900  IF ND% > IA% THEN ND% = IA%
```

```
9920  PRINT NME$(J)
9930  NEXT J
9940  GET CH$
9950  IF CH$ = R$ THEN 9970
9960  GOTO 9940
9970  IF ND% = IA% THEN 10010
9980  CALL  - 936
9990 STRT% = ND% + 1
10000  GOTO 9690
10010  POKE 34,6
10020  POKE 35,11
10030  CALL  - 936
10040 STRT% = 1
10050 ND% = STRT% + 59
10060  IF ND% >  LEN (TXT$) THEN
     ND% =  LEN (TXT$)
10070 ST$ =  MID$ (TXT$,STRT%,ND%
     )
10080  PRINT ST$
10090  GET CH$
10100  IF CH$ = R$ THEN 10120
10110  GOTO 10090
10120  IF ND% =  LEN (TXT$) THEN
     10160
10130  CALL  - 936
10140 STRT% = ND% + 1:ST$ = ""
10150  GOTO 10050
10160  POKE 34,11
10170  POKE 35,13
10180  CALL  - 936
10190  PRINT JRNL$
10200  POKE 34,13
10210  POKE 35,15
10220  CALL  - 936
10230  PRINT VVL$
10240  POKE 34,15
10250  POKE 35,17
10260  CALL  - 936
10270  PRINT PGS$
10280  POKE 34,17
10290  POKE 35,19
10300  CALL  - 936
10310  PRINT YR$
10320  IF IK% = 0 THEN 10460
10330  POKE 34,19: POKE 35,22
10340  CALL  - 936
10350 STRT% = 1
10360 ND% = STRT% + 1
10370  IF ND% > IK% THEN ND% = IK
     %
10380  FOR K = STRT% TO ND%
10390  PRINT CE$(K): NEXT K
10400  GET CH$
10410  IF CH$ = R$ THEN 10430
10420  GOTO 10400
10430  IF ND% = IK% THEN 10460
10440  CALL  - 936
10450 STRT% = ND% + 1: GOTO 10360

10460  TEXT : VTAB 23
10470  PRINT "ENTER KEY OR <RETUR
     N>"
10480  PRINT "IF NO CORRECTIONS"
10490  POKE 33,5
```

```
10520  POKE 35,23
10530  CALL  - 936
10540  INPUT KY$
10550  TEXT : HOME
10560  IF KY$ = "" THEN 8680
10570  VTAB 2
10580  IF KY$ = "A" THEN 10650
10590  IF KY$ = "T" THEN 10770
10600  IF KY$ = "J" THEN 10830
10610  IF KY$ = "V" THEN 10850
10620  IF KY$ = "P" THEN 10870
10630  IF KY$ = "R" THEN 10890
10640  IF KY$ = "K" THEN 10910
10650  PRINT "AUTHORS:": HTAB 1
10660  POKE 33,29
10670  POKE 32,11
10680  POKE 34,1
10690  POKE 35,10
10700  CALL  - 936
10710 IA% = 1
10720  INPUT NME$(IA%)
10730  IF NME$(IA%) = "" THEN 107
     50
10740 IA% = IA% + 1: GOTO 10720
10750 IA% = IA% - 1
10760  TEXT : GOTO 9630
10770  HOME : PRINT "TITLE:": HTAB
     1
10780  POKE 33,29
10790  POKE 32,11
10800  CALL  - 936
10810  INPUT TXT$
10820  TEXT : GOTO 9630
10830  INPUT "ENTER JOURNAL:";JRN
     L$
10840  GOTO 9630
10850  INPUT "ENTER VOLUME:";VVL$

10860  GOTO 9630
10870  INPUT "ENTER PAGES:";PGS$
10880  GOTO 9630
10890  INPUT "ENTER YEAR:";YR$
10900  GOTO 9630
10910 IK% = 1
10920  PRINT "ENTER KEYWORDS:": HTAB
     1
10930  POKE 33,29: POKE 32,14
10940  POKE 34,1: POKE 35,10
10950  CALL  - 936
10960  INPUT CE$(IK%)
10970  IF CE$(IK%) = "" THEN 1099
     0
10980 IK% = IK% + 1: GOTO 10960
10990 IK% = IK% - 1
11000  TEXT : GOTO 9630
11010  REM  STORE REF
11020  HOME : VTAB 5:RSP$ = ""
11030  INPUT "NEED TO CHECK THE D
     ATA?(Y/N)";RSP$
11040  IF RSP$ = "Y" THEN 9680
11050  HOME : VTAB 5:RSP$ = ""
11060  INPUT "VERIFY STORE! (Y/N)
     ";RSP$
11070  IF RSP$ = "Y" THEN 11100
```

```
11100  PRINT D$;"OPEN REFS,L255,D
     2"
11110  PRINT D$;"READ REFS,R0"
11120  INPUT ST$
11130 VOL$ =  LEFT$ (ST$,2)
11140  IF  VAL (VOL$) = DTA%(13) THEN
     11210
11150  PRINT D$;"CLOSE REFS"
11160  PRINT "VOLUME MISMATCH-INS
     ERT DISK R";DTA%(13);" INTO
     DRIVE 2, THEN PRESS <RETURN>
     "
11170  GET CH$
11180  IF CH$ = R$ THEN 11200
11190  GOTO 11170
11200  PRINT T$: GOTO 11100
11210  PRINT D$;"CLOSE REFS"
11220  PRINT D$;"OPEN REFS,L255,D
     2"
11230 IP% = 0:HP% = 0
11240 IP% = IP% + 1
11250  ON IP% GOTO 11260,11330,11
     340,11350,11360,11370,11410
11260  FOR K = 1 TO IA%
11270 HP% = HP% + 1
11280  IF K = IA% THEN 11300
11290 KP$(HP%) = NME$(K) + "&": GOTO
     11310
11300 KP$(HP%) = NME$(K) + "/"
11310  NEXT K
11320  GOTO 11240
11330 WRD$ = TXT$ + "/": GOTO 113
     80
11340 WRD$ = JRNL$ + "/": GOTO 11
     380
11350 WRD$ = VVL$ + "/": GOTO 113
     80
11360 WRD$ = PGS$ + "/": GOTO 113
     80
11370 WRD$ = YR$ + "#"
11380 HP% = HP% + 1
11390 KP$(HP%) = WRD$
11400  GOTO 11240
11410  HOME :BGN% = 0
11420  FOR K = 1 TO HP%
11430  IF ( LEN (BUFF$) +  LEN (K
     P$(K))) <  = 250 THEN 11460
11440 BGN% = K:K = HP%
11450  GOTO 11470
11460 BUFF$ = BUFF$ + KP$(K)
11470  NEXT K
11480 TTL$ =  STR$ ( LEN (BUFF$))

11490 XL% = 4:XP$ = TTL$:ST$ = ""

11500  GOSUB 30
11510 BUFF$ = ST$ + BUFF$
11520  REM  JOIN REF
11530 LBUFF% =  LEN (BUFF$)
11540 LREF% =  LEN (REF$)
11550 RN% = 255 - LREF%
11560  IF RN% = 0 THEN  GOTO 1164
     0
11570  IF LBUFF% > RN% THEN 11610
```

```
11590 BUFF$ = NLL$
11600  GOTO 11960
11610  REM  JOIN REF
11620 SME$ =  LEFT$ (BUFF$,RN%)
11630 REF$ = REF$ + SME$
11640  PRINT D$;"WRITE REFS,R";BN
     0%
11650  PRINT REF$
11660 BFLG% = 1
11670  IF BNO% < XND% THEN 11910
11680  REM  UPDT
11690 ST$ = "":XL% = 2:XP$ =  STR$
     (DTA%(14))
11700  GOSUB 30
11710 XL% = 3:XP$ =  STR$ (BNO%)
11720  GOSUB 30
11730  PRINT D$;"WRITE REFS,R0"
11740  PRINT ST$:ST$ = ""
11750  PRINT D$;"CLOSE REFS"
11760  HOME : PRINT "INSERT NEW D
     ISK IN DRIVE 2,THEN PRESS <R
     ETURN>"
11770  GET CH$
11780  IF CH$ = R$ THEN 11800
11790  GOTO 11770
11800  PRINT R$:BNO% = 1:DTA%(14)
      = DTA%(14) + 1
11810 VOL$ =  STR$ (DTA%(14)): PRINT
     "VOL=";VOL$
11820  PRINT D$;"OPEN REFS,L255,D
     2"
11830 ST$ = "":XL% = 2:XP$ = VOL$

11840  GOSUB 30
11850 VOL$ = ST$
11860 ST$ = ST$ + "001"
11870  PRINT D$;"WRITE REFS,R0"
11880  PRINT ST$:ST$ = ""
11890 DTA%(13) = DTA%(13) + 1
11900 DTA%(15) = BNO%:BFLG% = 0
11910  REM  LOAD REF WTH BUFF
11920 REF$ = NLL$
11930 RST% = LBUFF% - RN%
11940 REF$ =  RIGHT$ (BUFF$,RST%)

11950 BUFF$ = NLL$
11960  PRINT D$;"CLOSE REFS"
11970  IF BGN% = 0 THEN 12020
11980  FOR K = BGN% TO HP%
11990 BUFF$ = BUFF$ + KP$(K)
12000  NEXT K
12010 BGN% = 0
12020  FOR K = 1 TO HP%
12030 KP$(K) = ""
12040  NEXT K
12050 HP% = 0
12060  REM  AUTHR
12070  IF IA% = 0 THEN 12490
12080  HOME : VTAB 5: PRINT "INSE
     RT DISK A";DTA%(1);" INTO DR
     IVE 1, THEN PRESS <RETURN>"
12090  GET CH$: PRINT T$
12100 DV$(1) = "A" +  STR$ (DTA%(
     1))
```

```
12120  PRINT D$;"READ AWTBL,R0"
12130  INPUT ST$
12140 MDSKS% =  VAL ( LEFT$ (ST$,
     4))
12150  IF MDSKS% = DTA%(1) THEN 1
     2220
12160  PRINT D$;"CLOSE AWTBL"
12170  PRINT "VOLUME MISMATCH-INS
     ERT DISK A";DTA%(1);" INTO D
     RIVE 1, THEN PRESS <RETURN>"

12180  GET CH$
12190  IF CH$ = R$ THEN 12210
12200  GOTO 12180
12210  PRINT T$: GOTO 12110
12220  PRINT D$;"CLOSE AWTBL"
12230 FLNME$ = "AWTBL":RLN = 31:A
     FLG% = 1
12240  FOR J = 1 TO 6
12250  ON J GOTO 12260,12270,1228
     0,12290,12300,12310
12260 MDSKS% = DTA%(J): GOTO 1233
     0
12270 NREC% = DTA%(J): GOTO 12330

12280 NB% = DTA%(J): GOTO 12330
12290 P% = DTA%(J): GOTO 12330
12300 JPTR% = DTA%(J): GOTO 12330

12310  IF JPTR% = 0 THEN 12330
12320 H%(JPTR% - 1) = DTA%(J)
12330  NEXT J
12340 ST$ = "":XL% = 3:XP$ =  STR$
     (BNO%)
12350  GOSUB 30
12360 BCK$ = ST$
12370  GOSUB 3430
12380  FOR J = 1 TO 6
12390  ON J GOTO 12400,12410,1242
     0,12430,12440,12450
12400 DTA%(J) =  VAL ( MID$ (DV$(
     1),2)): GOTO 12480
12410 DTA%(J) = NREC%: GOTO 12480

12420 DTA%(J) = NB%: GOTO 12480
12430 DTA%(J) = P%: GOTO 12480
12440 DTA%(J) = JPTR%: GOTO 12480

12450  IF JPTR% = 0 THEN 12470
12460 DTA%(J) = H%(JPTR% - 1): GOTO
     12480
12470 DTA%(J) = 0
12480  NEXT J
12490  PRINT : IF IK% = 0 THEN 12
     960
12500  HOME : VTAB 5
12510  PRINT "INSERT DISK K";DTA%
     (7);"INTO DRIVE 1, THEN PRES
     S <RETURN>"
12520  GET CH$
12530  IF CH$ = R$ THEN 12550
12540  GOTO 12520
12550  PRINT T$:DV$(1) = "K" +  STR$
     (DTA%(7))
```

```
12570  PRINT D$;"READ KWTBL,R0"
12580  INPUT ST$
12590 MDSKS% =  VAL ( LEFT$ (ST$,
     4))
12600  IF MDSKS% = DTA%(7) THEN 1
     2670
12610  PRINT D$;"CLOSE KWTBL"
12620  PRINT "VOLUME MISMATCH-INS
     ERT DISK K";DTA%(7);" INTO D
     RIVE 1, THEN PRESS <RETURN>"

12630  GET CH$
12640  IF CH$ = R$ THEN 12660
12650  GOTO 12630
12660  PRINT T$: GOTO 12560
12670  PRINT D$;"CLOSE KWTBL"
12680 FLNME$ = "KWTBL":AFLG% = 2
12690  FOR J = 7 TO 12
12700 JJ = J - 6
12710  ON JJ GOTO 12720,12730,127
     40,12750,12760,12770
12720 MDSKS% = DTA%(J): GOTO 1279
     0
12730 NREC% = DTA%(J): GOTO 12790

12740 NB% = DTA%(J): GOTO 12790
12750 P% = DTA%(J): GOTO 12790
12760 JPTR% = DTA%(J): GOTO 12790

12770  IF JPTR% = 0 THEN 12790
12780 H%(JPTR% - 1) = DTA%(J)
12790  NEXT J
12800 ST$ = "":XL% = 3:XP$ =  STR$
     (BNO%)
12810  GOSUB 30
12820 BCK$ = ST$
12830  GOSUB 3430
12840  FOR J = 7 TO 12
12850 JJ = J - 6
12860  ON JJ GOTO 12870,12880,128
     90,12900,12910,12920
12870 DTA%(J) =  VAL ( MID$ (DV$(
     1),2)): GOTO 12950
12880 DTA%(J) = NREC%: GOTO 12950

12890 DTA%(J) = NB%: GOTO 12950
12900 DTA%(J) = P%: GOTO 12950
12910 DTA%(J) = JPTR%: GOTO 12950

12920  IF JPTR% = 0 THEN 12940
12930 DTA%(J) = H%(JPTR% - 1): GOTO
     12950
12940 DTA%(J) = 0
12950  NEXT J
12960  PRINT :X =  FRE (0)
12970  REM  UPDT
12980  IF BFLG% = 0 THEN 13090
12990  PRINT D$;"OPEN REFS,L255,D
     2"
13000 ST$ = "":XL% = 2:XP$ =  STR$
     (DTA%(14))
13010  GOSUB 30
13020 BNO% = BNO% + 1: PRINT "BNO
     =":BNO%
```

```
13050  PRINT D$;"WRITE REFS,R0"
13060  PRINT ST$:ST$ = ""
13070  PRINT D$;"CLOSE REFS"
13080 BFLG% = 0:DTA%(15) = DTA%(1
     5) + 1
13090 SFLG% = 1: GOTO 8680
13100  REM  MENU 3
13110  HOME : VTAB 5: PRINT "INSE
     RT DESK R";DTA%(13);" INTO D
     RIVE 2,THEN PRESS <RETURN>"
13120  GET CH$: PRINT T$
13130  PRINT D$;"OPEN REFS,L255,D
     2"
13140  IF REF$ = "" GOTO 13170
13150  PRINT D$;"WRITE REFS,R";BN
     0%
13160  PRINT REF$:REF$ = ""
13170  IF BUFF$ = "" GOTO 13210
13180 BN0% = BN0% + 1:DTA%(15) =
     DTA%(15) + 1
13190  PRINT D$;"WRITE REFS,R";BN
     0%
13200  PRINT BUFF$:BUFF$ = ""
13210  REM  UPDT REC0
13220 ST$ = "":XL% = 2:XP$ =  STR$
     (DTA%(14))
13230  GOSUB 30
13240 XL% = 3:XP$ =  STR$ (BN0%)
13250  GOSUB 30
13260  PRINT D$;"WRITE REFS,R0"
13270  PRINT ST$:ST$ = ""
13280  PRINT D$;"CLOSE REFS"
13290 JREC = 0
13300  HOME : VTAB 5: PRINT "SEAR
     CH REFERENCE:"
13310  VTAB 8
13320  PRINT "HELP"; SPC( 31);"H"

13330  VTAB 10
13340  PRINT "AUTHORS"; SPC( 28);
     "A"
13350  VTAB 12
13360  PRINT "KEYWORDS"; SPC( 27)
     ;"K"
13370  VTAB 14
13380  PRINT "PRINTOUT"; SPC( 27)
     ;"P"
13390  VTAB 16
13400  PRINT "QUIT"; SPC( 31);"Q"

13410  VTAB 22
13420  INPUT "ENTER KEY:";CY$
13430  IF CY$ = "H" THEN 13490
13440  IF CY$ = "A" THEN 13530
13450  IF CY$ = "K" THEN 13600
13460  IF CY$ = "P" THEN 13680
13470  IF CY$ = "Q" THEN 8320
13480  GOTO 13410
13490  REM   HELP SEARCH
13500 JB = 15:JE = 19: GOSUB 2840

13510  FOR K = 1 TO 2000: NEXT K
13520  GOTO 13300
13530  HOME : VTAB 5: PRINT "LIST
```

```
13540  INPUT NME$(IA%)
13550  IF NME$(IA%) = "" THEN 135
     70
13560 IA% = IA% + 1: GOTO 13540
13570 IA% = IA% - 1: HOME
13580  IF IA% = 0 THEN 13300
13590  GOSUB 6130:IA% = 0: GOTO 1
     3300
13600  HOME : VTAB 5: PRINT "LIST
      KEYWORDS-AT END,PRESS <RETU
     RN>":IK% = 1
13610  INPUT CE$(IK%)
13620  IF CE$(IK%) = "" THEN 1364
     0
13630 IK% = IK% + 1: GOTO 13610
13640 IK% = IK% - 1
13650  IF IK% = 0 THEN 13300
13660  GOSUB 6130:IK% = 0
13670  GOTO 13300
13680  HOME : VTAB 5
13690  IF X% > 0 THEN 13720
13700  PRINT "NO MATCHES FOUND"
13710  FOR JM = 1 TO 200: NEXT JM
     : GOTO 13300
13720  PRINT "SELECT PRINTOUT:"
13730  VTAB 10
13740  PRINT "IN ORDER OF REFEREN
     CES FOUND"; SPC( 7);"F"
13750  VTAB 12
13760  PRINT "USER ORDERED"; SPC(
     23);"U"
13770  VTAB 14
13780  PRINT "QUIT"; SPC( 31);"Q"

13790  VTAB 22
13800  INPUT "ENTER KEY:";CY$: HOME

13810  IF CY$ = "F" THEN 13850
13820  IF CY$ = "U" THEN 14960
13830  IF CY$ = "Q" THEN 13300
13840  GOTO 13790
13850  REM  AS FOUND
13860 LC% = 0:RLN = 255:JREC = 0:
     TMP$ = "":KREC% = 0:SW% = 0:
      HOME : VTAB 5
13870  PRINT "INSERT ANS DISK INT
     O DRIVE 1, THEN PRESS<RETURN
     >": GET CH$: PRINT R$: HOME

13880  PRINT "THE REFERENCES YOU
     REQUESTED": PRINT "ARE AS FO
     LLOWS:": PRINT
13890  PRINT D$;"OPEN ANS,L255,D1
     "
13900  FOR XY = 1 TO X%
13910 BUFF$ = NLL$:SW% = 0
13920 SX$ = ANS$(XY,2)
13930 AUTH$ =  LEFT$ (SX$,( LEN (
     SX$) - 5))
13940 LNG% =  LEN (AUTH$)
13950 DN% =  VAL ( MID$ (SX$,(LNG
     % + 1),2))
13960 IREC =  VAL ( RIGHT$ (SX$,3
     ))
```

```
13980  PRINT D$;"READ REFS,R0"
13990  INPUT ST$
14000 DK% =  VAL ( LEFT$ (ST$,2))

14010  IF DN% = DK% THEN 14080
14020  PRINT D$;"CLOSE REFS"
14030  PRINT "INSERT REFERENCE DI
     SK";DN%;" INTO DRIVE 2,THEN
     PRESS <RETURN>"
14040  GET CH$
14050  IF CH$ = R$ THEN 14070
14060  GOTO 14040
14070  PRINT T$: GOTO 13930
14080 DK% = DN%:ST$ = ""
14090  PRINT D$;"READ REFS,R";IRE
     C
14100  GOSUB 6050
14110 LG% =  LEN (ST$)
14120 STRT% = 1
14130  FOR IX = STRT% TO LG%
14140  IF  MID$ (ST$,IX,1) <  = "
     9" THEN 14170
14150  IF  MID$ (ST$,IX,LNG%) <  >
     AUTH$ THEN 14170
14160 LC% = IX:IX = LG%
14170  NEXT IX
14180  IF LC% = 0 THEN 14670
14190  REM  GETREF
14200 PLC% = LC% - 1:LC% = 0
14210  IF PLC% = 4 THEN 14260
14220  FOR JX = PLC% TO 1 STEP  -
     1
14230  IF  MID$ (ST$,JX,1) <  > "
     #" THEN 14250
14240 LC% = JX:JX = 1
14250  NEXT JX
14260 RECLN% =  VAL ( MID$ (ST$,(
     LC% + 1),4))
14270 RM% =  LEN (ST$) - LC%
14280  IF RM% < RECLN% THEN 14470

14290 BUFF$ =  MID$ (ST$,(LC% + 5
     ),RECLN% - 1)
14300 ZFLG% = 0:ZT$ = NLL$
14310  IF JREC = 0 THEN 14400
14320  FOR YZ = 1 TO JREC
14330  PRINT D$;"READ ANS,R";YZ
14340  INPUT ZT$
14350  IF ZT$ <  > BUFF$ THEN 143
     70
14360 ZFLG% = 1:YZ = JREC
14370  NEXT YZ
14380  IF ZFLG% = 0 THEN 14400
14390  GOTO 14430
14400 JREC = JREC + 1
14410  PRINT D$;"WRITE ANS,R";JRE
     C
14420  PRINT BUFF$
14430 TMP% = LC%
14440 LC% =  LEN (BUFF$) + 5 + TM
     P%
14450  IF LC% = LG% THEN 14670
14460 BUFF$ = NLL$:STRT% = LC%:LC
     % = 0: GOTO 14130
```

```
14490 BUFF$ =  MID$ (ST$,(LC% + 5
     ),(RM% - 4))
14500 IREC = IREC + 1:ST$ = ""
14510  PRINT D$;"READ REFS,R";IRE
     C
14520  GOSUB 6050
14530 SX$ =  LEFT$ (ST$,(RECLN% -
      LEN (BUFF$) - 1))
14540 BUFF$ = BUFF$ + SX$
14550 ZFLG% = 0:ZT$ = NLL$
14560  FOR YZ = 1 TO JREC
14570  PRINT D$;"READ ANS,R";YZ
14580  INPUT ZT$
14590  IF ZT$ <  > BUFF$ THEN 146
     10
14600 ZFLG% = 1:YZ = JREC
14610  NEXT YZ
14620  IF ZFLG% = 0 THEN 14640
14630  GOTO 14670
14640 JREC = JREC + 1
14650  PRINT D$;"WRITE ANS,R";JRE
     C
14660  PRINT BUFF$:BUFF$ = ""
14670  PRINT D$;"CLOSE REFS"
14680  NEXT XY
14690  IF JREC = 0 THEN 14750
14700 ST$ = NLL$
14710  FOR YZ = 1 TO JREC
14720  PRINT D$;"READ ANS,R";YZ
14730  INPUT ST$: PRINT YZ;":";ST
     $
14740  NEXT YZ
14750  PRINT D$;"CLOSE ANS"
14760  PRINT : PRINT "TO CONTINUE
     , PRESS ANY KEY"
14770  GET CH$: PRINT T$
14780  HOME : VTAB 5: PRINT "SEAR
     CH OPTIONS:"
14790  VTAB 8: PRINT "NEW FILE FO
     R": VTAB 9: PRINT "ANY MATCH
     ES FOUND"; SPC( 10);"N"
14800  VTAB 11: PRINT "ADD TO EXI
     STING": VTAB 12: PRINT "FILE
      OF MATCHES"; SPC( 12);"A"
14810  VTAB 14: PRINT "ALTERNATE
     PRINTOUT"; SPC( 9);"L"
14820  VTAB 16: PRINT "QUIT"; SPC(
     23);"Q"
14830  VTAB 22: INPUT "ENTER KEY:
     ";CY$
14840  IF CY$ = "N" THEN 14930
14850  IF CY$ = "A" THEN 14950
14860  IF CY$ = "L" THEN 14960
14870  IF CY$ = "Q" THEN 14890
14880  GOTO 14820
14890 RSP$ = NLL$: HOME : VTAB 5:
      INPUT "VERIFY QUIT! (Y/N)";
     RSP$
14900  IF RSP$ = "Y" THEN 14950
14910  IF RSP$ = "N" THEN 14760
14920  GOTO 14890
14930  FOR XY = 1 TO X%:ANS$(XY,1
     ) = "":ANS$(XY,2) = "": NEXT
     XY
```

```
14960  HOME : VTAB 5
14970  IF JREC > 0 THEN 14990
14980  PRINT : PRINT "SELECT OPTI
     ON F, THEN U": GOTO 13680
14990 N% = 0
15000  INPUT "ENTER RECNO OR A ZE
     RO:";RC%
15010  IF RC% = 0 THEN 15050
15020 N% = N% + 1
15030 NU%(N%) = RC%
15040  GOTO 15000
15050  HOME : PRINT "INSERT ANS D
     ISK INTO DRIVE 1, THEN PRESS
      <RETURN>"
15060  GET CH$: IF CH$ = R$ THEN
     15070: GOTO 15050
15070  PRINT R$: PRINT D$;"OPEN A
     NS,L255,D1"
15080  FOR K = 1 TO N%
15090 IREC% = NU%(K):ST$ = ""
15100  PRINT D$;"READ ANS,R";IREC
     %
15110  INPUT ST$: PRINT ST$
15120  NEXT K
15130  PRINT D$;"CLOSE ANS"
15140  FOR K = 1 TO N%:NU%(K) = 0
     : NEXT K
15150  HOME : GOTO 13300
15160  REM  QUIT
15170  HOME
15180  IF SFLG% = 1 THEN 15200
15190  GOTO 15800
15200  REM  SAVE PTRS
15210  IF REF$ = "" THEN 15250
15220  PRINT D$;"OPEN REFS,L255,D
     2"
15230  PRINT D$;"WRITE REFS,R";BN
     0%
15240  PRINT REF$
15250  IF BUFF$ = "" THEN 15290
15260 BN0% = BN0% + 1:DTA%(15) =
     DTA%(15) + 1
15270  PRINT D$;"WRITE REFS,R";BN
     0%
15280  PRINT BUFF$:BUFF$ = ""
15290 ST$ = "":XL% = 2:XP$ =  STR$
     (DTA%(14))
15300  GOSUB 30
15310 XL% = 3:XP$ =  STR$ (BN0%)
15320  GOSUB 30
15330  PRINT D$;"WRITE REFS,R0": PRINT
     ST$:SX$ = ST$:ST$ = ""
15340  PRINT D$;"CLOSE REFS"
15350  HOME : VTAB (5)
15360  PRINT "INSERT SYSTEM MASTE
     R DISK INTO DRIVE 1, THEN PR
     ESS <RETURN>"
15370  GET CH$
15380  IF CH$ = R$ THEN 15400
15390  GOTO 15370
15400  PRINT R$: PRINT D$;"OPEN I
     NDXPTRS,L31,D1"
15410 IREC = 0:ST$ = "":XL% = 4
15420 STRT% = 1:ND% = 6
```

```
15450  GOSUB 30
15460  NEXT K
15470  PRINT D$;"WRITE INDXPTRS,R
     ";IREC
15480  PRINT ST$
15490  PRINT D$;"CLOSE INDXPTRS"
15500  IF IREC = 1 THEN 15600
15510  PRINT "INSERT AWTBL1 INTO
     DRIVE 2, THEN PRESS <RETURN>
     "
15520  GET CH$
15530  IF CH$ = R$ THEN 15550
15540  GOTO 15520
15550  PRINT T$: PRINT D$;"OPEN A
     WTBL,L31,D2"
15560  PRINT D$;"WRITE AWTBL,R0"
15570  PRINT ST$:ST$ = ""
15580  PRINT D$;"CLOSE AWTBL"
15590  GOTO 15680
15600  PRINT "INSERT KWTBL1 INTO
     DRIVE 2, THEN PRESS <RETURN>
     "
15610  GET CH$
15620  IF CH$ = R$ THEN 15640
15630  GOTO 15610
15640  PRINT T$: PRINT D$;"OPEN K
     WTBL,L31,D2"
15650  PRINT D$;"WRITE KWTBL,R0"
15660  PRINT ST$:ST$ = ""
15670  PRINT D$;"CLOSE KWTBL"
15680 IREC = IREC + 1
15690  IF IREC = 2 THEN 15730
15700 STRT% = 7:ND% = 12
15710  PRINT D$;"OPEN INDXPTRS,L3
     1,D1"
15720 ST$ = "": GOTO 15430
15730 ST$ = "":XP$ =  STR$ (DTA%(
     13))
15740  GOSUB 30
15750 ST$ = ST$ + SX$
15760  PRINT D$;"OPEN INDXPTRS,L3
     1,D1"
15770  PRINT D$;"WRITE INDXPTRS,R
     ";IREC
15780  PRINT ST$:ST$ = ""
15790  PRINT D$;"CLOSE INDXPTRS"
15800  END

]
```

BIBLIOGRAF

USER'S GUIDE

Table of Contents

I.Index

II.Sample Job Displays

USER'S GUIDE

1.Sign on

-insert system master diskette into drive 1 and close the door

-turn on the monitor

-turn on the printer

-power on keyboard.

When these steps are carried out, the system will be 'booted'. That is, the programs necessary for running the system and reading and writing to disk will be stored in the computer memory.

2.Format diskettes

The diskettes that are used to store information pertaining to the bibliographic references must be formatted to accept the data. Several diskettes may be formatted at one time so that there will be spares available should a reference diskette become full during the course of storing information on it. In order to format a diskette,use the following method:

-insert the system master diskette into drive 1 and type "in#6 <return>"

-type "load hello" and wait for the program to be loaded into the computer - the blinking cursor will appear on the

screen

-insert a blank diskette into drive 1 and type "in#6<return>"

-type "init hello" - when the whirring and clacking noise from the drive has subsided, the diskette will have been formatted for future use. Be sure to label it with the following information - DOS 3.3/48K system - so that you know it has been formatted already.

-in order to format several diskettes, follow the last two steps for each one.

3.Create a new Index file

This program is used to create a new index file of pointers to author,keyword or reference files ONLY WHEN A NEW REFERENCE SYSTEM IS CREATED. It should never be run after a reference system has been started because valuable and irreplaceable information will have been lost. To create a new index file of pointers, use the following method:

-insert the system master diskette into drive 1

-type "run crindx" and a screen message will appear as follows:

Is this a new file?(y/n)

-if yes, then type "y"<return>. The file will be

initialized and the system returned to the user.

-if no, the system will printout another screen message as follows:

Caution - your file of ptrs to the author,keyword and reference files is about to be deleted,then reinitialized -(y/n)?

-if yes, then type"y"<return>and the file will be reinitialized as stated

-if no,then type"n"<return>and the system is returned to the user for further processing.

4.Initialize an Author,Keyword or Reference file

This program is used to initialize an author, keyword or master reference file, each on a separate diskette.
The file name and length of each record is written on the file directory so that it is ready to accept information when a new reference is stored. To perform this operation, do the following:

-insert the system master diskette into drive 1

-type "run initfl" and a screen message will appear:

Enter filetype(0/1/2):

-the user should enter in the number corresponding to the file he wishes to initialize, that is,

0)quit the program - the system is returned to the user

1)author or Keyword file: the user will be prompted to enter in specific information about the file, i.e, filename, record length, drive no. and number of records per file. This information is:

Awtbl,31,2 or

Kwtbl,31,2

and

3935<return>.

The system will prompt the user to insert the appropriate disk into drive 2 before initialization begins.

2)reference file - the user will be prompted to insert a diskette into drive 2 that will be prepared to accept reference information on it.

-To insure that there are always diskettes available should a data diskette become full during the course of a store operation, the user should initialize several diskettes of each type as described above when a new bibliographic system is begun and thereafter periodically depending on how much data is stored per session.

CAUTION - if there are no initialized diskettes available when needed during a store operation, the current input data will be stored only partially and therefore the files and index pointers will become erroneous. Therefore make sure that there are enough initialized diskettes available before each session to avoid causing havoc with the information already in the reference system.

5. Backup files - A MUST

Files containing information on author, keyword, full references or pointers to the individual files must be backed up before a new session begins. That is, a copy of each file is made to the appropriate backup disk before any additional information is entered into the system. The backup files can then be used to restore the files should a power failure or any other disaster destroy the data on the working diskettes during an input session.

Thus in order to copy a diskette, follow these steps:

-have several initialized diskettes on hand

-load the system master diskette into drive 1, type "in#6"<return> and wait for the blinking cursor to appear on the screen

-type "brun fid"<return> and wait for the program to be loaded - a screen display will show the various options available to the user.

-to copy a file, type "1"<return>

-type in the following information after the system prompts as shown below:

```
      Source slot? 6
            drive? 1
 Destination slot? 6
            drive? 2
         Filename? Awtbl or
                   Kwtbl or
                   Refs
```

-the system will ask you to insert the appropriate disks to proceed with the copy or press <esc> key to return to the main menu.

-if you proceed with the copy and the diskette in drive 2 already has a file with the copy name, you have a choice of renaming the transferred file, overwriting the file already stored on the diskette in drive 2 or aborting the copy task by pressing the <esc> key.

-follow the system prompts to return to the main menu after the copy has been performed.

-to quit the session select "9"<return>and you will be returned to the monitor and the blinking cursor for alternate routing.

6.Store or Search for a reference

-Insert the system master diskette into drive 1 and close door.

-tupe "new" - this command will clear the computer of any previous programs that were loaded into it.

-Type "run biblio" - this is the main program to enter or search for a bibliographic reference. The system will prompt the user to insert a specific diskette at the appropriate time or to select a routing method. An on-line help facility will also be available should the user require additional information on a particular activity.

-Note: When this program is run, a printout will be produced simultaneously.

MENU DISPLAYS

Menu 1

Help	H
Enter reference	E
Search reference	S
Quit	Q
Enter option	

This display is presented when you first load the system or when you have finished with a particular function in which case you are returned to this menu by the system.

Options selected

H - will give you a brief introduction to the overall bibliographic system and what you can do with it.

-when "H" is entered, a screen message will be displayed to insert the system master diskette into drive 1, then press <return>.

-an overview of the bibliographic system will be scrolled on the screen. When you wish to return to Menu 1, press the <return> key.

E - will allow you to enter in reference information. You will be guided by the system from Menu 2 which will allow you to input, check or store your data. You will then

be returned to this menu.

-when this option is chosen, disk load instructions will be printed on the screen to insert the proper author diskette into drive 1 and the proper reference diskette into drive 2, followed by a <return>. The system will display Menu 2 for further processing.

S - will allow you to search for a reference in the master files working from a third menu, that is, Menu 3. You will be prompted by the system to enter in either author names or keywords which will form the basis of the search. If any matches are found, the location of the references will be stored on a temporary file. You will be given the option of narrowing the search utilizing this information or be returned to the Menu 1 for alternate routing.

Q - will allow you to terminate the session but only after certain pointers are saved in a special file. The system will prompt you to insert the system master diskette into drive 1 and the author or keyword file diskette into drive 2 so that the current index pointers may be stored on the master index files. After this housekeeping has been done, the session will end.

-remove the diskettes from both drives and label them with a soft felt tip (only) please. Make sure that there are enough formatted diskettes available for backup or to

use in the next session. If not, follow the guidelines that were previously given to format diskettes.

-turn off the monitor

-turn off the printer

-power off the keyboard.

Menu 2

Enter reference:

Help	H
Input	I
Check	C
Store	S
Quit	Q
Enter key	

This menu will allow you to either enter in a new reference or once entered, to check the data before storing it on file. You will be returned to Menu 1.

Options selected

H - will serve as a guide for using the options selected above. When "H" is entered, a screen message to insert the system master diskette into drive 1 will be displayed. After this has been done, press <return> and a description of the Enter reference operation will be scrolled by on the screen. When you wish to continue processing, press <return> and you will be returned to Menu 2.

I - will enable you to enter information for a new reference. The system will prompt you to enter in author names, title, journal, volume, pages, year of publication

and keywords. You will then be returned to Menu 2.

-when "I" is entered, a list of the headings as noted above will be displayed on the screen. The system will guide you to enter in data for each heading at the blinking cursor. Thus, to facilitate data entry,

-use the left arrow key to backspace over any typing errors if possible

-enter the data followed by the <return> key

-if no more data, enter <return> only and the cursor will proceed to the next heading.

-if the length of the title is greater than that allowed by the system (max 255 characters), the computer will beep when the limit has been exceeded. In this case, use the Check facility to reenter a shortened version of the title.

-at end of data input, the system will give you a chance to check your input data as described in the next section.

C - will enable you to check your input data for errors before storing it on file. The system will display all the data including author's names, title,journal, volume, year of publication, pages and keywords. If any corrections are to be made, then the system will guide you to reenter a specific item, as noted above. If you wish to reenter all the reference information, then return to Menu 2 and select the "Input" option again. Remember that errors are much

more difficult to repair once the information has been stored on the master reference file - they should be corrected using the Check facility before proceeding to the "Store" option.

Thus the screen message that is displayed 1)after the data has been input, 2)when the "C" option has been selected from Menu 2 and 3)when the "S" option has been selected from Menu 2 is:

Checkdata (y/n)?

-if "y" is entered, the current input data will be printed out adjacent to the proper subheadings. If there is more data than available space on the screen for a particular subheading, e.g., author, you must manually scroll the data in groups by using the <return> key after each group has been presented. When all the data has been printed out on the screen, you enter in the letter corresponding to the heading to be corrected followed by a <return> key or just the <return> key if there are no corrections to be made.

-if "n" is entered, you will be returned to Menu 2 if you were in the data input mode or the "C" option or moved on for further processing if you had just selected the "S" option.

S - once the input data has been entered, the system will store the information in the appropriate place in the master reference file. Before this task is accomplished, however, you will have another chance to check the current input data for the last time before the reference is stored on file. When the "Store" option is selected, you will be presented with the "Checkdata(y/n)?" message.

-if "y" is chosen, the system will enable you to check and correct your data as described above.

-if "n" is selected, then another screen message will be displayed. You must verify the "Store" operation before it is initiated,i.e.,

Verify store!(y/n)

-enter "y" or "n" then <return>

-if "n", you will be returned to Menu 2 for further processing

-if "y", then instructions to insert the proper author diskette into drive 1 will be displayed on the screen. Having done this task, press <return> and the system will proceed to store the current input data. During the course of the store operation, the system will print out the author name, its storage address and the current index pointers. This information should be saved so that it can be used as an aid in reconstructing the file should a malfunction occur

in the system.

After the author information has been stored, the keyword information is then processed. A screen message to insert the proper keyword diskette into drive 1 will be displayed on the screen. Thereafter all keywords pertaining to this reference will be stored on a separate keyword index file. Again, the keywords and file index pointers will be printed out as the operation takes place. Save the information to use in an audit trail if necessary.
When all the data has been stored, you will be returned to Menu 2.

Q - will allow you to return to the main menu, either for alternate routing or to end the session.

-if you select "Q", the system will present a screen display for you to verify quitting, i.e.,

Verify quit!(y/n).

-enter "y" or "n" then <return>

-if "n" ,you will be returned to Menu 2

-if "y", you will be directed to Menu 1 to select another activity or to quit the system entirely.

Menu 3

Search reference:

Help	H
Authors	A
Keywords	K
Printout	P
Quit	Q
Enter key	

This menu allows you to search for a reference based on authors or keywords. If the search is successful, then you will be able to printout the full reference from the master file. If there is more than one master diskette, then the system will prompt you to insert the correct diskette into a specific drive.

Options

H - should serve as a guide to performing a search of the reference file and obtaining a printout thereafter.

-if "H" is selected, a message will be displayed on the screen to insert the system master diskette into drive 1, then press <return>. When this has been done, a descriptive guide will be printed out on the screen. You should insert the previous disk into drive 1, then press <return> to return to Menu 3.

A - will allow you to enter one or several author names as the basis of the search. The author index file in drive 1 will be checked for any matches. If there is a match, the location in the master reference file will be saved for later use.

-when this option is chosen, you will be asked to list the author names that will form the basis of the search.

-enter in each name at the blinking cursor , followed by a <return>

-if there are no entries, just enter <return>

-during the course of the search operation, the system will printout instructions to insert the appropriate diskette into drives 1 and 2, including a diskette to store the results of a successful search. This diskette should be labelled the 'Ans' diskette for printout purposes. After the search has been completed, you will be returned to Menu 3.

K - will allow you to enter one or several keywords as the basis of the search. The keyword index file in drive 1 will be checked for a match, and if successful, then the location in the master reference file will be saved for further processing. Basically, the same procedure as for author names is followed.

P - will allow you to printout the disk and record number of a match or the full reference from the master file itself.

-when this option is selected, you will be presented with Menu 4 that gives you the choice of how you want the references printed out or the option to quit.

Menu 4

Print instructions:

As found	F
User ordered	U
Quit	Q

Enter key

-enter in the appropriate key and then press <return>

Options

F - will printout the references as they were found in the search. The system will instruct you to insert the "Ans" diskette into drive 1. For the first search, this diskette should be a new formatted one.

-once the references have been listed, the system will pause and display a message, i.e.,

To continue, press any key

-when you are ready to continue on the system, enter a key and Menu 5 will be displayed on the screen where you will have further options for additional searches.

U - will allow you to rearrange the printout list once an initial printout of references as found in the search has been obtained. The system will instruct you to enter in the numbers of the references from the previous printout in the new order in which you wish the references to be printed.

Note: you must obtain a listing of the references found in the search before you can reorder the printout.

Q - returns you to Menu 3 for further processing.

Menu 5

Additional searches:

New file	N
Add to file	A
Alternate printout	L
Quit	Q
Enter key	

Options

N - any matches found in the next search will constitute a new listing. You will be returned to Menu 3.

A - references found in the next search will be added to those found in previous searches. You will be returned to Menu 3.

L - option to printout the references in a different order. The system will ask you to type in the record numbers of the references from the previous printout in the new order in which you now wish them printed.

Q - you will be returned to Menu 3 after verifying the quit step, i.e.,

-if yes, then you will be returned to Menu 3

-if no, then you will be returned to Menu 5 to try again.

SAMPLE JOB DISPLAYS

```
 RUN CRINDX
IS THIS A NEW FILE?(Y/N)Y
FILE INITIALIZED
```

```
]RUN
IS THIS A NEW FILE?(Y/N)N
CAUTION-YOUR FILE OF PTRS TO THE AUTHOR,KEYWORD AND MASTER REFERENCE FILES IS AB
OUT TO BE DELETED,THEN REINITIALIZED -- (Y/N)?Y
FILE INITIALIZED
```

```
]RUN
IS THIS A NEW FILE?(Y/N)N
CAUTION-YOUR FILE OF PTRS TO THE AUTHOR,KEYWORD AND MASTER REFERENCE FILES IS AB
OUT TO BE DELETED,THEN REINITIALIZED -- (Y/N)?N

]
```

```
 RUN INITFL,D1
ENTER FILETYPE - (0/1/2):1
ENTER FLNME,RECLEN,DRVNO:AWTBL,31,2
ENTER NO RECORDS:3900
INSERT DISK INTO DRIVE 2THEN PRESS <RETURN>


DO YOU WANT A LISTING?(Y/N)N
ENTER FILETYPE - (0/1/2):2
INSERT REFS DISK INTO DRIVE 2,THEN PRESS <RETURN>



]
```

```
 RUN
ENTER FILE NAME:
?DESCRIP
ENTER REC LENGTH:
?254
ENTER DRIVE NO:2
ENTER STRT REC NO:
?1
ENTER END REC NO:
?19
INSERT DISK INTO DRIVE 2, THEN PRESS <RETURN>


1:
WELCOME TO BIBLIOGRAF - A SYSTEM FOR STORING AND RETRIEVING BIBLIOGRPHIC REFEREN
CES ON A MICROCOMPUTER.  THE MAIN FUNCTIONS ON THIS SYSTEM INCLUDE STORAGE, RETR
IEVAL AND PRINTING OF INFORMATION FOR A REFERENCE INCLUDING AUTHOR NAMES,TITLE,
2:
JOURNAL, YEAR OF PUBLICATION AND INCLUSIVE PAGES.  SINCE THIS SYSTEM HAS BEEN DE
SIGNED FOR BOTH THE NOVICE AND EXPERT USER, YOU WILL BE GUIDED BY THE SYSTEM AS
MUCH AS POSSIBLE THROUGH THE USE OF MENUS DISPLAYED ON THE SCREEN.  FROM THESE
3:
MENUS, YOU CAN SELECT THE FUNCTIONS TO BE PERFORMED BUT YOU WILL ALWAYS BE ABLE
TO REFER TO THE HELP FACILITY FOR INFORMATION ON A SPECIFIC FUNCTION BEFORE PROC
EEDING TO IT.  IF YOU CHOOSE TO END A SESSION, SELECT THE <QUIT> OPTION.  THE
4:
SYSTEM WILL VERIFY THAT YOU DO INDEED WISH TO END THE SESSION AND THEN PERFORM C
ERTAIN HOUSEKEEPING TASKS BEFORE LETTING YOU GO.  THESE TASKS INVOLVE STORING CE
RTAIN INFORMATION ON FILE ON SPECIFIC DISKS WHICH YOU WILL BE ASKED TO INSERT AT
5:
THE APPROPRIATE TIME IN THE PROPER DRIVE.  SO - GOOD LUCK AND DON'T FORGET TO CC
PY YOUR DATA FILES ONTO A BACKUP DISK BEFORE THE START OF EACH SESSION.
6:
YOU HAVE ELECTED TO ENTER REFERENCE DATA INTO THE COMPUTER FOR SUBSEQUENT STORAG
E ON THE MASTER REFERENCE FILE.  INSERT THE SYSTEM MASTER DISKETTE INTO DRIVE 1
AND THE DISKETTE CONTAINING THE MOST RECENTLY STORED REFERENCES IN DRIVE 2.
7:
IF THERE IS A NEED FOR A DIFFERENT DISKETTE TO BE IN EITHER DRIVE, THE SYSTEM WI
LL PROMPT YOU AT THE APPROPRIATE TIME.  TO INPUT DATA FOR A REFERENCE, FOLLOW TH
E SAMPLE FORM GIVEN IN THE APPENDIX.  TYPE IN THE REQUIRED INFORMATION FOLLOWED
8:
BY A <RETURN> KEY.  IF NO DATA IS TO FOLLOW, JUST ENTER THE <RETURN> KEY.  NOTE
THAT THE FORMAT FOR AUTHOR NAMES IS FIRST INITIAL+PERIOD+LAST NAME WITH NO INTER
VENING COMMAS OR SPACES.  IF THERE IS MORE THAN ONE AUTHOR, THE SYSTEM WILL
9:
PROMPT YOU TO ENTER THE OTHER NAMES AFTER THE BLINKING CURSOR.
10:
THE CHECK FACILITY IS PROVIDED SO THAT YOU CAN CHECK THE DATA JUST ENTERED FOR A
NY ERRORS PRIOR TO STORING IT ON THE MASTER FILE.  THIS CHECK IS IMPORTANT SINCE
 REFERENCE INFORMATION IS NOT EASILY CORRECTED ONCE IT IS STORED ON FILE.  YOU
11:
CAN CORRECT AS MANY ITEMS AS NECESSARY, BUT IF YOU WISH TO REENTER ALL THE INFOR
MATION, THEN SELECT THE <INPUT> OPTION RATHER THAN REENTERING THE DATA, ONE BY O
NE.
12:
THE STORE FUNCTION WILL STORE THE DATA THAT YOU HAVE JUST ENTERED ONTO THE MASTE
R REFERENCE FILE IN DISK 2.  IN ADDITION THE AUTHOR NAMES WILL BE STORED IN AN A
UTHOR INDEX FILE AND SUBSEQUENTLY THE KEYWORDS WILL BE STORED ON A KEYWORD INDEX
13:
FILE.  THE SYSTEM WILL PROMPT YOU TO INSERT THE CORRECT DISKETTE INTO DRIVE 1.
14:
THE QUIT OPTION RETURNS YOU TO THE MAIN MENU FOR ALTERNATE ROUTING.
15:
```

F YOU WISH TO SEARCH FOR A PARTICULAR AUTHOR OR AUTHORS, SELECT THE A OPTION AND
 THE SYSTEM WILL GUIDE YOU TO INPUT THE NAMES. OTHERWISE SELECT THE K OPTION
16:
AND YOU CAN ENTER IN THE KEYWORDS THAT WILL FORM THE BASIS OF YOUR SEARCH.
17:
IF THE SEARCH HAS BEEN SUCCESSFUL YOU WILL BE ABLE TO PRINTOUT ANY OR ALL OF THE
 REFERENCES THAT WERE LOCATED. SPECIFICALLY, YOU CAN OBTAIN A PRINTOUT IN THE O
RDER IN WHICH THE REFERENCES WERE FOUND OR YOU CAN SELECTIVELY ORDER THEM YOURSE
LF.
18:
IF YOU CHOOSE THE LATTER OPTION, THE SYSTEM WILL PROMPT YOU TO ENTER IN THE NUMB
ER OF THE REFERENCE IN THE ORDER OF PRINTOUT.
19:
THE QUIT OPTION ALLOWS YOU TO CHOOSE ANOTHER FUNCTION OR TO END THE SESSION.

]

```
 RUN BIBLIO,D1

          BIBLIOGRAPHIC STORAGE
                    AND
            RETRIEVAL SYSTEM
SYSTEM OPTIONS:
SYSTEM DESCRIPTION                    D
ENTER REFERENCE                       E
SEARCH REFERENCE                      S
QUIT                                  Q
ENTER KEY:E
INSERT DISK A1 INTO DRIVE 1
INSERT DISK R1 INTO DRIVE 2, THEN PRESS <RETURN>


ENTER REFERENCE:
HELP                                  H
INPUT                                 I
CHECK DATA                            C
STORE                                 S
QUIT                                  Q
ENTER KEY:I
 AUTHORS:
 TITLE:
 JOURNAL:
 VOLUME:
 YEAR:
 PAGES:
?D.FRANKS
?P.HAMET
?J.PLAMONODONDON
?H.CHIU
?
?TESTING1111111111111111111111111111111111111111111111111111111111111111111
?JJJ
?11
?1980
?1-10
KEYWORDS:
?
BEFORE YOU STORE THIS DATA ON FILE, YOU SHOULD CHECK IT FOR ENTRY ERRORS.
CHECKDATA (Y/N)?Y
CHECK OF INPUT DATA:                 KEY
AUTHORS:                              A
TITLE:                                T
JOURNAL:                              J
VOLUME:                               V
PAGES:                                P
YEAR:                                 R
KEYWORDS:                             K
D.FRANKS
P.HAMET
J.PLAMONDON
H.CHIU
TESTING1111111111111111111111111111111111111111111111111111111111111111111
111111111111
JJJ
11
1-10
1980
ENTER KEY OR <RETURN>
IF NO CORRECTIONS
?
```

```
INPUT                                   I
CHECK DATA                              C
STORE                                   S
QUIT                                    Q
ENTER KEY:S
NEED TO CHECK THE DATA?(Y/N)N
VERIFY STORE! (Y/N)Y
INSERT DISK A1 INTO DRIVE 1, THEN PRESS <RETURN>

1:D.FRANKS
ADR=0
2:P.HAMET
ADR=0
3:J.PLAMONDON
ADR=0
4:H.CHIU
ADR=0
P,J:0             1
H(J),H(J-1):2     1


1
H.CHIU                  010010000
2
P.HAMET                 010010000
3
00000000000000000000000000000
4
J.PLAMONDON             010010000
5
D.FRANKS                010010000
6
00000000000000000000000000000



ENTER REFERENCE:
HELP                                    H
INPUT                                   I
CHECK DATA                              C
STORE                                   S
QUIT                                    Q
ENTER KEY:I
 AUTHORS:
 TITLE:
 JOURNAL:
 VOLUME:
 YEAR:
 PAGES:
?D.FRANKS
?H.CHIU
?J.DOE
?S.KIRBY
?
?TESTING22222222222222222222222222222222222222222222222222222222222222222222222222222
222
?KKK
?22
?1982
?2-20
KEYWORDS:
?
BEFORE YOU STORE THIS DATA ON FILE, YOU SHOULD CHECK IT FOR ENTRY ERRORS.
CHECKDATA (Y/N)?N
```

```
INPUT                                   I
CHECK DATA                              C
STORE                                   S
QUIT                                    Q
ENTER KEY:S
NEED TO CHECK THE DATA?(Y/N)Y
CHECK OF INPUT DATA:                   KEY
AUTHORS:                                A
TITLE:                                  T
JOURNAL:                                J
VOLUME:                                 V
PAGES:                                  P
YEAR:                                   R
KEYWORDS:                               K
D.FRANKS
H.CHIU
J.DOE
S.KIRBY
TESTING22222222222222222222222222222222222222222222222222222?
22222222222222222222
KKK
22
2-20
1982
ENTER KEY OR <RETURN>
IF NO CORRECTIONS
?
ENTER REFERENCE:
HELP                                    H
INPUT                                   I
CHECK DATA                              C
STORE                                   S
QUIT                                    Q
ENTER KEY:S
NEED TO CHECK THE DATA?(Y/N)N
VERIFY STORE! (Y/N)Y
INSERT DISK A1 INTO DRIVE 1, THEN PRESS <RETURN>

1:D.FRANKS
ADR=1
P,J:0            1
H(J),H(J-1):2    1

2:H.CHIU
ADR=0
P,J:0            1
H(J),H(J-1):2    1

3:J.DOE
ADR=0
P,J:1            2
H(J),H(J-1):3    2

4:S.KIRBY
ADR=1
P,J:0            2
H(J),H(J-1):4    3


1
J.DOE                010010000
2
H.CHIU               010010000
3
```

```
D.FRANKS                010010000
5
000000000000000000000000000000
6
000000000000000000000000000000
7
H.CHIU                  010010000
8
S.KIRBY                 010010000
9
D.FRANKS                010010104
10
J.PLAMONDON             010010000
11
000000000000000000000000000000
12
000000000000000000000000000000


BNO=2
ENTER REFERENCE:
HELP                                    H
INPUT                                   I
CHECK DATA                              C
STORE                                   S
QUIT                                    Q
ENTER KEY:I
 AUTHORS:
 TITLE:
 JOURNAL:
 VOLUME:
 YEAR:
 PAGES:
?P.HAMET
?J.PLAMONDON
?D.FRANKS
?
?TESTING33333333333333333333333333333333333333333333333333333333333333333333333333333
333
?LLL
?132
?1982
?3-30
KEYWORDS:
?
BEFORE YOU STORE THIS DATA ON FILE, YOU SHOULD CHECK IT FOR ENTRY ERRORS.
CHECKDATA (Y/N)?N
ENTER REFERENCE:
HELP                                    H
INPUT                                   I
CHECK DATA                              C
STORE                                   S
QUIT                                    Q
ENTER KEY:C
BEFORE YOU STORE THIS DATA ON FILE, YOU SHOULD CHECK IT FOR ENTRY ERRORS.
CHECKDATA (Y/N)?Y
CHECK OF INPUT DATA:                   KEY
AUTHORS:                                A
TITLE:                                  T
JOURNAL:                                J
VOLUME:                                 V
PAGES:                                  P
YEAR:                                   R
```

```
J.PLAMONDON
D.FRANKS
TESTING333333333333333333333333333333333333333333333333333
33333333333333333333
LLL
132
3-30
1982
ENTER KEY OR <RETURN>
IF NO CORRECTIONS
?
ENTER REFERENCE:
HELP                                H
INPUT                               I
CHECK DATA                          C
STORE                               S
QUIT                                Q
ENTER KEY:S
NEED TO CHECK THE DATA?(Y/N)N
VERIFY STORE! (Y/N)Y
INSERT DISK A1 INTO DRIVE 1, THEN PRESS <RETURN>

1:P.HAMET
ADR=0
P,J:0             2
H(J),H(J-1):4     3

2:J.PLAMONDON
ADR=3
P,J:0             2
H(J),H(J-1):4     3

3:D.FRANKS
ADR=2
P,J:1             3
H(J),H(J-1):5     4


1
P.HAMET                 010020000
2
H.CHIU                  010010000
3
P.HAMET                 010010114
4
D.FRANKS                010010113
5
J.DOE                   010010000
6
00000000000000000000000000000
7
00000000000000000000000000000
8
S.KIRBY                 010010000
9
D.FRANKS                010010104
10
J.PLAMONDON             010010000
11
00000000000000000000000000000
12
J.PLAMONDON             010020000
13
D.FRANKS                010020000
```

```
15
000000000000000000000000000000

ENTER REFERENCE:
HELP                                H
INPUT                               I
CHECK DATA                          C
STORE                               S
QUIT                                Q
ENTER KEY:Q
VERIFY QUIT! (Y/N)Y
SYSTEM OPTIONS:
SYSTEM DESCRIPTION                  D
ENTER REFERENCE                     E
SEARCH REFERENCE                    S
QUIT                                Q
ENTER KEY:Q
VERIFY QUIT!(Y/N)Y
INSERT SYSTEM MASTER DISK INTO DRIVE 1, THEN PRESS <RETURN>

INSERT AWTBL1 INTO DRIVE 2, THEN PRESS <RETURN>

INSERT KWTBL1 INTO DRIVE 2, THEN PRESS <RETURN>

]
```

```
]RUN BIBLIO,D1

INSERT DISK R1 INTO DRIVE 2, THEN PRESS <RETURN>


          BIBLIOGRAPHIC STORAGE
                    AND
            RETRIEVAL SYSTEM
SYSTEM OPTIONS:
SYSTEM DESCRIPTION                    D
ENTER REFERENCE                       E
SEARCH REFERENCE                      S
QUIT                                  Q
ENTER KEY:S
INSERT DESK R1 INTO DRIVE 2,THEN PRESS <RETURN>

SEARCH REFERENCE:
HELP                                  H
AUTHORS                               A
KEYWORDS                              K
PRINTOUT                              P
QUIT                                  Q
ENTER KEY:A
LIST AUTHORS-AT END,PRESS <RETURN>
?D.FRANKS
-
1:D.FRANKS
INSERT AWTBL DISK 1 INTO DRIVE 1, THEN PRESS <RETURN>

SEARCH REFERENCE:
HELP                                  H
AUTHORS                               A
KEYWORDS                              K
PRINTOUT                              P
QUIT                                  Q
ENTER KEY:P
SELECT PRINTOUT:
IN ORDER OF REFERENCES FOUND          F
USER ORDERED                          U
QUIT                                  Q
ENTER KEY:F
INSERT ANS DISK INTO DRIVE 1, THEN PRESS<RETURN>


THE REFERENCES YOU REQUESTED
ARE AS FOLLOWS:




1:D.FRANKS&P.HAMET&J.PLAMONDON&H.CHIU/TESTING1111111111111111111111111111111111
1111111111111111111111111111111111111111111111111111111111111111111111111/JJJ/11/1-10/1
1
2:D.FRANKS&H.CHIU&J.DOE&S.KIRBY/TESTING2222222222222222222222222222222222222222
22222222222222222222222222222222222222222222222222222222222222/KKK/22/2-20/1982

TO CONTINUE, PRESS ANY KEY

SEARCH OPTIONS:
```

```
ADD TO EXISTING
FILE OF MATCHES              A
ALTERNATE PRINTOUT           L
QUIT                         Q
ENTER KEY:N
SEARCH REFERENCE:
HELP                              H
AUTHORS                           A
KEYWORDS                          K
PRINTOUT                          P
QUIT                              Q
ENTER KEY:A
LIST AUTHORS-AT END,PRESS <RETURN>
?J.PLAMONDON
?
1:J.PLAMONDON
INSERT AWTBL DISK 1 INTO DRIVE 1, THEN PRESS <RETURN>

SEARCH REFERENCE:
HELP                              H
AUTHORS                           A
KEYWORDS                          K
PRINTOUT                          P
QUIT                              Q
ENTER KEY:P
SELECT PRINTOUT:
IN ORDER OF REFERENCES FOUND      F
USER ORDERED                      U
QUIT                              Q
ENTER KEY:F
INSERT ANS DISK INTO DRIVE 1, THEN PRESS<RETURN>


THE REFERENCES YOU REQUESTED
ARE AS FOLLOWS:



1:D.FRANKS&P.HAMET&J.PLAMONDON&H.CHIU/TESTING1111111111111111111111111111111111
1111111111111111111111111111111111111111111111111111111111111111111/JJJ/11/1-10/1
1
2:P.HAMET&J.PLAMONDON/TESTING3333333333333333333333333333333333333333333333333
33333333333333333333333333/LLL/33/3-30/1983

TO CONTINUE, PRESS ANY KEY

SEARCH OPTIONS:
NEW FILE FOR
ANY MATCHES FOUND            N
ADD TO EXISTING
FILE OF MATCHES              A
ALTERNATE PRINTOUT           L
QUIT                         Q
ENTER KEY:L
ENTER RECNO OR A ZERO:2
ENTER RECNO OR A ZERO:1
ENTER RECNO OR A ZERO:0
INSERT ANS DISK INTO DRIVE 1, THEN PRESS <RETURN>


P.HAMET&J.PLAMONDON/TESTING3333333333333333333333333333333333333333333333333
33333333333333333333333333/LLL/33/3-30/1983
D.FRANKS&P.HAMET&J.PLAMONDON&H.CHIU/TESTING1111111111111111111111111111111111
1111111111111111111111111111111111111111111111111111111111111111111/JJJ/11/1-10/1981
SEARCH REFERENCE:
```

```
AUTHORS                                A
KEYWORDS                               K
PRINTOUT                               P
QUIT                                   Q
ENTER KEY:Q
SYSTEM OPTIONS:
SYSTEM DESCRIPTION                     D
ENTER REFERENCE                        E
SEARCH REFERENCE                       S
QUIT                                   Q
ENTER KEY:Q
VERIFY QUIT!(Y/N)Y

]
```