

SHARE PROGRAM LIBRARY AGENCY

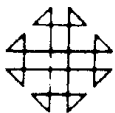


PROGRAM NUMBER

067018

University of Miami

1365 MEMORIAL DRIVE - CORAL GABLES, FLORIDA
(305) - 284-6257



CONTRIBUTED PROGRAM LIBRARY SUBMITTAL FORM
(for IBM S/360, 1130 and 1800)

SHARE Program Library Agency
Triangle Universities Computation Center
P. O. Box 12076
Research Triangle Park, N. C. 27709

This form should be completed and submitted with the program package to PID at the address shown above. Standards and instructions for submitting programs are in your *User Group Reference Manual* or the *Contributed Program Submittal Standards Manual* available from PID.

- ① Program Order Number (to be filled in by PID) 360D-06.7.018
- ② System Type (machine) S / 3 6 0
- ③ Search Key R A N D O M A C C E S S B I N A R Y S E
A R C H O F S E Q U E N T I A L F I L E S
O N D I S K
- ④ Name of Author (if different than submitter's)
- ⑤ Submitter's Name (direct technical inquiries to) Ronald K. Sipherd
- ⑥ Submitter's Address The Boeing Co., Comm. Airplane Div.
Computing Department
P. O. Box 707
Renton, Washington 98055
- ⑦ Title of Program BSEARCH - A Random Access Binary-Search Technique
for Sequential Files on Disk or Drum
- ⑧ Submitter's User Group Affiliation Code and Installation Code S B R
- ⑨ Submitter's Own Program Identification and Suffix (optional) B S C H
- ⑩ Primary Subject Code 0 6 . 7
- ⑪ Secondary Subject Codes
- ⑫ Operating or Monitor System Required S / 3 6 0
- ⑬ New or Revision Code (if revision, show prior Program Order Number, in item 1) N
- ⑭ Year Completed 6 8
- ⑮ Date of Submittal 1 2 : 6 8
- ⑯ Documentation (number of original pages submitted) 7
- ⑰ Abstract (should contain sufficient information for a reader to determine the value of the program). Listed on the reverse side of this form are subjects which may serve as a guide for a descriptive abstract.

Subject Guide

- ## ABSTRACT

(Please attach additional pages if necessary) Total pages attached 1

"I hereby give anyone permission to reprint, reproduce, and distribute this program to anyone else."

(18) Signature of Submitter and Date

(19) Signature of Installation Addressee

T4SF

CONTRIBUTED PROGRAM LIBRARY SUBMITTAL FORM

Subject Guide

- a. Purpose
- b. Programming Language used
- c. Version and modification level or release number of IBM Programming System used, or program order number for non-IBM authored program used
- d. Field of application
- e. Type of routine (main program, subroutine, etc.)
- f. Specific description of machine requirements
- g. Engineering Changes (EC) level of equipment (if pertinent)

ABSTRACT

attachments for optimization techniques.

Notes: Currently implemented only for 2311, 2314, and 2301 devices.

DISCLAIMER

~~Triangle Universities Computation Center (TUCC)~~
serves solely as the distribution agent for contributed programs and does not test or maintain them. They are distributed essentially in the original form submitted by the author. Neither TUCC nor SHARE, INC., makes any warranty, expressed or implied, as to the documentation, function, or performance of the contributed programs.

(Please attach additional pages if necessary) Total pages attached _____

Permission to Publish

"I hereby give anyone permission to reprint, reproduce, and distribute this program to anyone else."

18 Signature of Submitter and Date _____

19 Signature of Installation Addressee _____

T4SF

TABLE OF CONTENTS

Deck Key	Page	
	4	DECK KEY
Purpose	5	
Method	5	
Usage (Operating Procedures) for BAL Program for COBOL Program	6 7 7	Deck #1 - Source deck, sequence 00000100 through 00048100 in c.c. 73-80; 481 cards.
Error Codes	9	Deck #2 - Object deck, sequence 0001-0035 in c.c. 77-80; BSCH in c.c. 73-76; 35 cards.
Subroutines Called	9	
Precision	9	
Restrictions	9	
Checkout Results	9	
Flowcharts	10	

SUBJECT:

AUTHOR:

PURPOSE:

COBOL Subroutine BSEARCH

R. K. Sipherd

To search a sequentially-organized file on a direct-access device containing fixed length records, blocked or unblocked, by means of a binary-search technique. The desired logical record is placed into a user-specified receiving area if it can be found; if it is not found, blanks are placed into the area.

METHOD:

After entering the routine, saving registers, etc., the program takes one of two paths: If this is not the first time the routine has been entered, so that core has already been obtained for the buffer and various internal parameters set, it initializes only those parameters that may have been previously altered and goes directly into the search routine proper. If the routine is being entered for the first time (or for the first time following execution of "BSCHCLSE"), it must execute the primary-setup phase.

This phase, after altering a branch so as to bypass itself in future calls, performs the following functions:

1. Opens an internal DCB, using the DDNAME passed to it in the caller's DCB.
2. Obtains the TTR (relative disk address) of the last block in the file from the DSCB of the search-file. An I/O error reading the DSCB causes an abend, user code 025.
3. Calculates from the blocksize passed in the caller's DCB, the number of blocks per track for the file, and abends (user code 050) if the unit does not occur in a list of acceptable device types, or if (user code 075) the blocksize is greater than the track capacity.
4. Calculates the number of blocks in the file.
5. Obtains main storage for an input buffer; amount of storage equals file blocksize.

6. Initializes other internal parameters, such as setting the length-byte of the "compare" instructions to equal the key-length.

Control then passes to the search routine itself, which does the following:

1. Calculates the number of the block to be read, and converts that value to an absolute disk address via nucleus-resident routine IEPCNVT.
2. Performs I/O using the XDAP macro, and checks for successful completion.
3. If I/O unsuccessful, writes console message and prepares to fill receiving area with X's, then goes to record-not-found routine.
4. If I/O successful, goes to record-compare routine.
5. If record-key matches passed-key, goes to record-found output routine; otherwise, resets internal values preparatory to re-read operation and goes back to #1, unless record read is the first or the last in the file, in which case it branches to the record-not-found routine.

The record-found routine moves the selected record into the user-specified area, restores the registers, and returns control the calling program.

The record-not-found routine fills the user-specified area with a string of blanks or X's, restores the registers, and returns control to the calling program.

The second entry-point to the program, BSCHCLSE, is used when it is desired to search a different file. The routine closes the internal DCB, frees the storage allocated for the search-file buffer, and returns.

USAGE:

I. Parameter Requirements

- A. 'CALL' statement requires five parameters, in this order:

1. DCB (or FD-NAME in COBOL).
2. Key to be searched for.
3. Length of the key (binary full-word).
4. Position of the high-order byte of the key within the record (binary full-word).
5. Area into which is to be placed:
 - a. The record (if key-match found).
 - b. Blanks (if no matching record found).
 - c. X's (if I/O error occurs while reading).

II. Usage Via Assembler Language

A. In assembler language, calling format would be:

:
:
: CALL ESEARCH, (MYDCB, MYKEY, KEYLENG,
POSITN, MYAREA)

```
CALL BSCHLSE
***Optional--closes EXCP
DCB and releases buffer
storage***
.
```

```
MYDCB DCB DORG=PS, MACRF=G, DDNAME=
SEARCHME, BLKSIZE=1020, LRECL=102
```

MYKEY DC C'1234567890ABCDEF GH'

KEYLENG DC F'18' ***length of 'MYKEY'***

POSITN DC F'85' ***key starts at position
85 of record***

```

MYAREA DS CL102
***area for record,
      blanks, or X's--must be
      .GE. LRECL, or
      following area of
      memory may be overlaid***

```

III. Usage Via COBOL

DATA DIVISION.

FILE SECTION.

-7-

```
FD MY-DCB
RECORDING MODE IS F
BLOCK CONTAINS 10 RECORDS
RECORD CONTAINS 102 CHARACTERS
LABEL RECORDS ARE STANDARD
DATA RECORD IS MY-RECORD.

001 MY-RECORD.
:
WORKING-STORAGE SECTION.

777 MY-KEY PICTURE IS X(18)
VALUE IS 'ABCDE9876543210c*$.

777 LENGTH-OF-MY-KEY
PICTURE IS S999999
USAGE IS COMPUTATIONAL
VALUE IS 18.

777 KEY-STARTS-AT
PICTURE IS S999999
USAGE IS COMPUTATIONAL
VALUE IS 40.

001 RECEIVING-AREA.
:
: *****Size of area must be
: *****Define subfields of
PROCEDURE DIVISION.
:
: MOVE DESIRED-CHARGE-NUMBER TO MY-KEY.
ENTER LINKAGE.
CALL 'BSEARCH' USING MY-DCB, MY-KEY, LE
RECEIVING-AREA.
ENTER COBOL.
:
ENTER LINKAGE.
CALL 'BSCHCLSE'.
ENTER COBOL.
:
: Optional---f
enables use
```

— 8 —

ERRORS:

Abend user code 0025: Error in reading DSCB.
Contents of Register 5 (in binary):

04 - Disk not mounted.
08 - DSCB not on VTOC
12 - I/O error in DSCB

Abend user code 0050: File is not on an
acceptable unit (acceptable units are 2311,
2314, and 2301).

Abend user code 0075: Blocksize is greater than
capacity of a track.

Receiving area filled with blanks: No record
exists in the file with that key.

Receiving area filled with X's: I/O error
while trying to read the search-file.

SUBROUTINES CALLED: None

PRECISION: Not applicable.

RESTRICTIONS:

I. Avoid:

- A. Track-overflow.**
- B. Variable-length or undefined records.*
- C. Use of devices (for search-file) other
than 2311, 2314, or 2301. (Additional
direct-access devices may be included by
adding 9-10 cards in the BSEARCH source
deck).
- D. Incorrect parameters (error-detection
routines omitted to save memory, results
can be wildly unpredictable).
- E. Searching a file which contains no records;
every read-attempt will come back as an I/O
error.**

II. The DCB for the file to be searched must be closed
at the time BSEARCH is called, and must specify
DDNAME, blocksize, and logical-record length
(cannot be supplied via JCL).

CHECKOUT RESULTS:

Working from a standard file of about 5,600 logical
records 102 characters long, the routine was
tested with various blocking factors, single and
multiple disk extents, on a Model 65 and a Model
75-50 attached support processor. The routine
appears to be error-free as long as the specified
constraints are adhered to.

*Has not been tested, assumed not to work.

**Has been tested.

CHART TITLE - BINARY SEARCH ROUTINE FOR FIXED-LENGTH SAM FILES ON DISK, BY R.X.

```

      / BSEARCH /
      |
      | NOTE 01
      | * * * * *
      | CSECT *
      | * * * * *
      |
      | P12 02
      |-----|
      | SAVE |
      | [14,12], |
      | BSEARCH-ENTRY |
      |-----|
      |
      |
      | 03
      |-----|
      | LR R12,R15 |
      | SET UP BASE |
      | REGISTER |
      |-----|
      |
      |
      | 04
      |-----|
      | ST |
      | F13,SAVEAREA+4 |
      | |
      | LR R10,R13 |
      | |
      | LA |
      | R13,SAVEAREA |
      | |
      | ST R13,B(R10) |
      |-----|
      |
      |
      | 05
      |-----|
      | LR R10,R1 |
      | USE R10 FOR PARAM |
      | REGISTER |
      |-----|
      |
      | STEP#1 06
      | ...
      | : 2.01:
      | : ... INITIAT

```

CHART TITLE - BINARY SEARCH ROUTINE FOR FIXED-LENGTH SAM FILES ON DISK, BY R&K.

**
PRIMARY SET-UP
*

(1.0A---)*
INITIAL I 01

MVI I
RTBRN1+1,K100
SET RTBRN1 TO A
NC-CP

L R8,C(010)
LOAD ADDR OF DCB
INTO R.3

I
I
I 02

MVC
XDCB+4C(8),
DCBDDNAM
MOVE DCB DDNAME
INTO XDCB DCB

I
I
I 03

OPEN (XDCB)
OPEN XDCB FILE

ICBTAINI ROUTINE

I
I
I 04

L R9,XDCB+44
LOAD DCB ADDR
INTO R.9

L R7,32(R9)
LOAD DCB ADDR
INTO R.7

SH R9,R1131

I
I
I 05

MVC
DCBCHRR+0(R9)

MVC
DCBOLSER+28(R7)

I
I
I 06

GET STORAGE FOR
DCB

I
I
I 07

OBTAIN LRP&M
READ DCB IN

I NOTE 09

CH R15,R104
TEST RETURN CODE
IF RETCODE LESS
THAN 4, ALL IS
WELL

I
I
I 09

SEE NOTE
ABOVE

I
I
I 10

IN/LOW

I
I
I 11

L R,R15

I
I
I 11

ARENDALL

BEND
25,0000,00000000
ELSE BEND WITH DOW
MESSAGE

I
I
I 12

D30K

L R1,TRMKADDR
LOAD DSCB ADDR
INTO R.1

MVC
DSILSTAR(1,
981R1)
STORE LAST-BLOCK
PTR

I
I
I 13

FREEMAIN
R,LV=350,A=111
FREE DSCB STORAGE
AREA

I
I
I 14

SR R9,R9

I
I
I 15

CLC
18(2,R7),LITX2001
IS IT A 2311
IF NOT, GO TO
CHECK IF 2314

I
I
I 16

SEE NOTE
ABOVE

I
I
I 16

EQUAL

I
I
I 16

BEND

BEND
25,0000,00000000
ELSE BEND WITH DOW
MESSAGE

UCBTST2 I NOTE 17

CLC
18(2,R7),LITX2008
IS IT A 2314
IF NOT, GO TO
CHECK IF 2301

I
I
I 18

SEE NOTE
ABOVE

I
I
I 18

UNEQL

I
I
I 18

EQUAL

I
I
I 18

UNEQL

BEND
25,0000,00000000
ELSE BEND WITH DOW
MESSAGE

CHART TITLE - BINARY SEARCH ROUTINE FOR FIXED-LENGTH SAM FILES ON DISK, BY R.L.K.

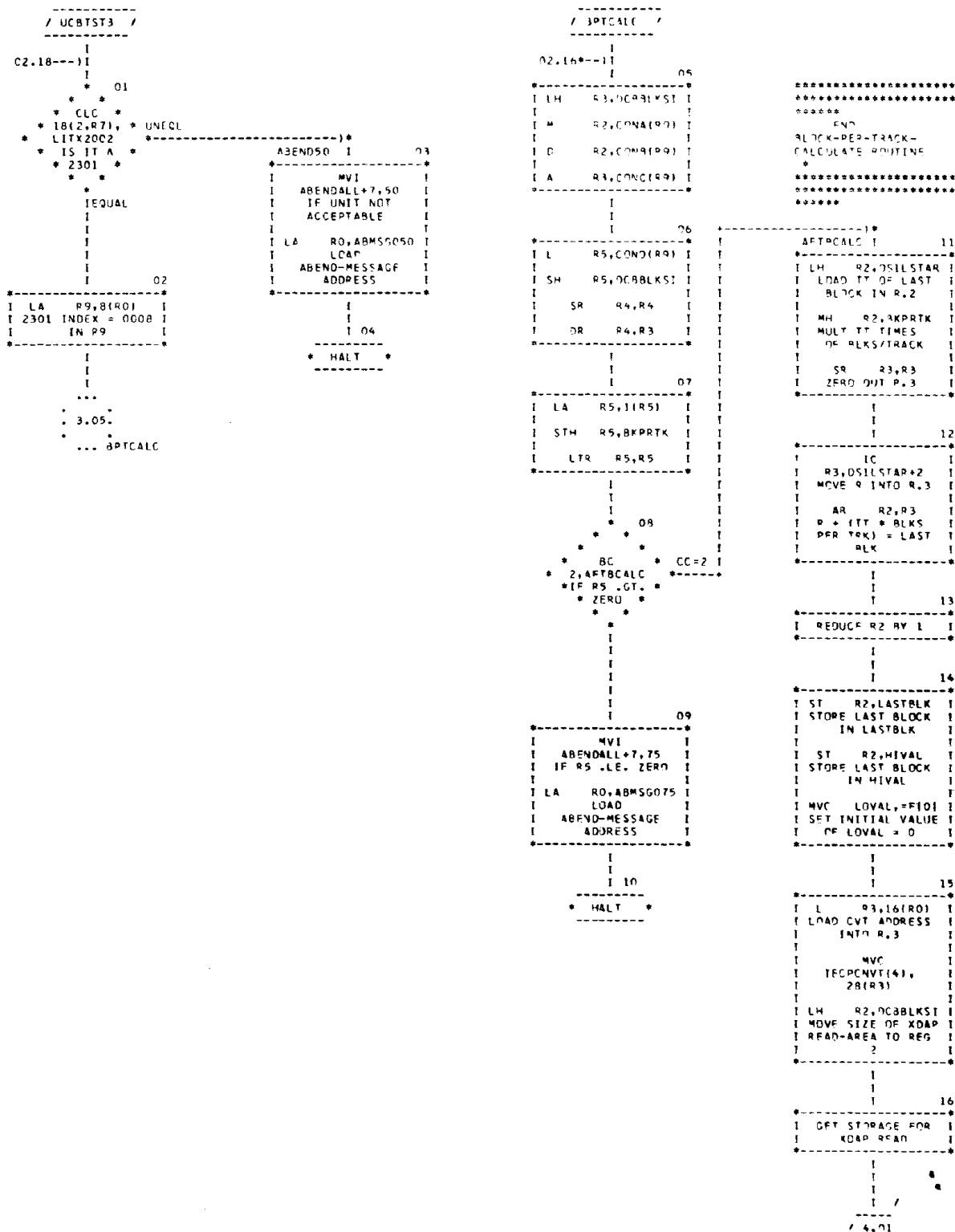


CHART TITLE - BINARY SEARCH ROUTINE FOR FIXED-LENGTH SAM FILES ON DISK, BY R.K.

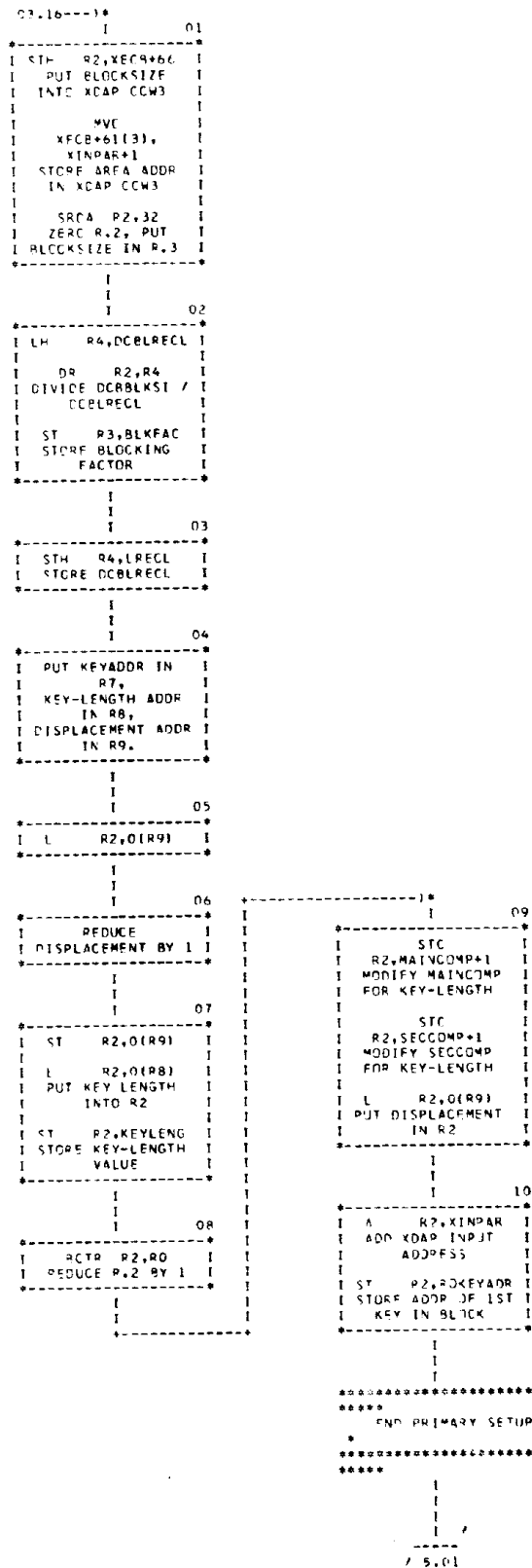


CHART TITLE - BINARY SEARCH ROUTINE FOR FIXED-LENGTH SAM FILES ON DISK, BY R.K.

C4,10---)*

 THIS IS THE BINARY
 SEARCH ROUTINE
 PROPER. THE PREVIOUS
 ROUTINES *
 HAVE INITIALIZED THE
 COUNTERS AND OPENED
 THE FILE TO BE
 SEARCHED.*

BLKCALC 1 01
 L R2,HIVAL
 S R2,LOVAL
 LA R2,1(R2)
 ADD 1 TO R2 FOR
 UPWARD ROUNDING

02
 SRA R2,1
 DIVIDE R2 BY 2
 A R2,LOVAL
 ADD 1/2 OF
 DIFFERENCE TO
 LOVAL
 ST R2,BKND
 STORE RESULT IN
 BKND

 (PERFORM MODULE
 ARITHMETIC + CHG BKND
 TO REL. TRK NO)

03
 LH R4,BKPRK
 LOAD DIVISOR INTO
 R4
 L R2,BKND
 LOAD DIVIDEND
 INTO R2
 SRDA R2,32
 ZERO R2, OUT
 DIVIDEND INTO R3

04
 DR R2,R4
 BKND / BKPRK
 LA R2,1(R2)
 ADD 1 TO REGNUM
 (XR)
 STH R3,XTT
 STORE QUOTIENT IN
 XTT

05
 STC R2,XP
 STORE REMAINDER
 IN XP

 LINK TO
 RELATIVE-TRACK-TO-
 ABSOLUTE-TRACK
 CONVERSION
 ROUTINE (IEPCNVT)

06
 L R0,XTTPN
 L R1,XDCB+44
 LA R2,XMBCHR
 STH R9,R13,SAVE913
 SAVE REGISTERS

07
 LP R3,R12
 REG 3 TEMPORARILY
 BECOMES BASE
 REGISTER

08
 L R15,IEPCNVT

09
 (REG. VALUE)
 BALR P14,R15
 BRANCH-AND-
 LINK TO
 IEPCNVT
 ROUTINE

10
 LH R9,R13,SAVE913
 RESTORE REGISTERS

11
 LA R2,XDAPRETN
 LOAD RETURN
 ADDRESS INTO R2

 XDAP-AND-ERROR-CHECK
 SUBROUTINE

12
 XDAP
 XECB,R1,XDCB,
 XINPAR,0025,
 XMBCHR
 READ A BLOCK

13
 * PAUSE *

WAIT ECB=XECB

NOTE 14
 * * * * *
 * CLC XECB,X17FI
 * TEST ECB FOR I/O
 * ERROR
 * RETURN TO PGM IF
 * I/O O.K.
 * * * * *

15
 * * * * *
 * SEE NOTE
 * ABOVE
 * * * * *

16
 * * * * *
 * XDAPRETN
 * * * * *

NOTE 16
 * * * * *
 * CLC
 * BKND, LASTBLK
 * IF NOT, TEST
 * FOR (TRUNCATED)
 * LAST-BLOCK
 * IF IT IS LAST
 * BLOCK, ALL IS
 * WELL
 * * * * *

17
 * * * * *
 * SEE NOTE
 * ABOVE
 * * * * *

18
 * * * * *
 * XDAPRETN
 * * * * *

NOTE 18
 * * * * *
 * *** I/O ERROR IN
 * BINARY-SEARCH
 * ROUTINE
 * ***
 * * * * *

19
 * * * * *
 * SEE NOTE ABOVE
 * * * * *

20
 * * * * *
 * MVI 0
 * FORSET+1,CHG
 * TO FILL PCVG AREA
 * WITH XIS
 * * * * *

7.03
 ... NOTFOUND

-15-

CHART TITLE - BINARY SEARCH ROUTINE FOR FIXED-LENGTH SAM FILES ON DISK, BY P.K.

[illegible]

CHART TITLE - BINARY SEARCH ROUTINE FOR FIXED-LENGTH SAM FILES ON DISK, BY R.X.

