

Datashare 3 Course Outline

Day 1

- A. Datashare or Databus 1100 Hardware Configuration & Capabilities
 - B. Interpretive Language-DBCMP
 - C. Data Definition DBCMP User's Guide pg. 3-2
 - 1. DIM page 3-2
 - 2. FORM page 3-2
 - 3. INIT page 3-2
 - D. Keyboard I/O
 - 1. KEYIN page 7-1
 - 2. DISPLAY page 7-7
 - E. Arithmetic Commands
 - 1. ADD page 6-2
 - 2. SUB page 6-2
 - 3. COMPARE page 6-3
 - 4. MOVE page 6-3
 - 5. Flags page 6-1
 - F. Transfer of Control
 - 1. GOTO page 4-1
 - 2. STOP page 4-3
 - G. Sequential Disk I/O page 7-10,7-16
 - 1. File Formatting, FILE statement page 3-1
 - 2. PREPARE, OPEN, CLOSE page 7-18,7-20,7-21
 - 3. Sequential READ, WRITE, WEOF page 7-25,7-33,7-39
 - H. Generating, Compiling & Running a Datashare program page 8-1
- LAB: Sequential File Processing
READING ASSIGNMENT: DOS. Manual-EDIT page 13-1
DBCMP Manual-Arithmetic Instructions
and Keyboard I/O pages 7-1/7-7

Day 2

A. Arithmetics (continued)

- 1. MULT page 6-2
- 2. DIV page 6-3
- 3. STORE page 6-4
- 4. LOAD page 6-4

B. Transfer of Control

- 1. BRANCH page 4-1
- 2. CALL, RETURN page 4-2
- 3. CHAIN page 4-3
- 4. BEEP page 7-8

C. CONSOLE, CLOCK page 7-7,5-9

D. Disk

- 1. Hardware
- 2. Random File Format & Design page 7-16
- 3. Random READ, WRITE, WEOF page 7-25,7-33
- 4. Continuation Read & Write page 7-25,7-26
- 5. READ tab, WRITAB page 7-26,7-35

E. DOS COMMANDS DOS USER'S GUIDE

- CAT,CHANGE,COPY,EDIT,FREE,KILL
- LIST,MIN,MOUT,NAME,REWIND,SAPP

LAB: Random File Processing

READING ASSIGNMENT: DOS. User's Guide (PART III, Review Commands covered in class)

Day 3

- A. PRINT, RELEASE page 7-8, 7-10
- B. ROLLOUT page 4-7
- C. DOS. CHAIN COMMAND DOS USER'S GUIDE
- D. String Instructions pages 5-1/5-12
 - 1. Formpointer & Logical Length
 - 2. Instructions to change FP & LL
BUMP, RESET, ENDSET, LENSET
CLEAR, EXTEND
 - 3. Data Manipulation Instructions
CMATCH, CMOVE, TYPE, MOVE
APPEND, LOAD, STORE
- E. INCLUDE, EQU page 8-5, 8-6

LAB: String Instructions

READING ASSIGNMENT:

DBCMP USER'S GUIDE-STRING INSTRUCTION page 5-1/5-12

Day 4

- A. TRAP & Trappable Conditions page 4-4
- B. Disk (continued)
 - 1. Indexed File Format page 7-16
 - 2. Indexed READ, WRITE, READKS,
UPDATE, INSERT, DELETE page 7-27/7-31
page 7-34,7-36/7-38
- C. DOS Commands DOS USER'S GUIDE
 - 1. REFORMAT
 - 2. INDEX
- D. DOS Tables DOS.USER'S GUIDE
- E. DOS Commands
 - 1. Files
 - 2. DUMP

LAB: Indexed Sequential File Processing

READING ASSIGNMENT: DBCMP User's Guide-Disk I/O

Day 5

A. ANSWER & MASTER

B. Language Extensions (DB11 and 5500 DS)

1. KEYIN - *IT, *JL, *JR, *ZF, *DE
2. WRITE - *ZF, *MP
3. REPLACE & SEARCH

C. Datashare 3 System Considerations

1. Keyboard & Screen I/O
2. Disk I/O
3. Paging, TABPAGE page 4-10
4. File Contention & PI page 4-8

D. DOS Generation (including DOSGEN,BOOTMAKE)

E. Putting DS on Disk

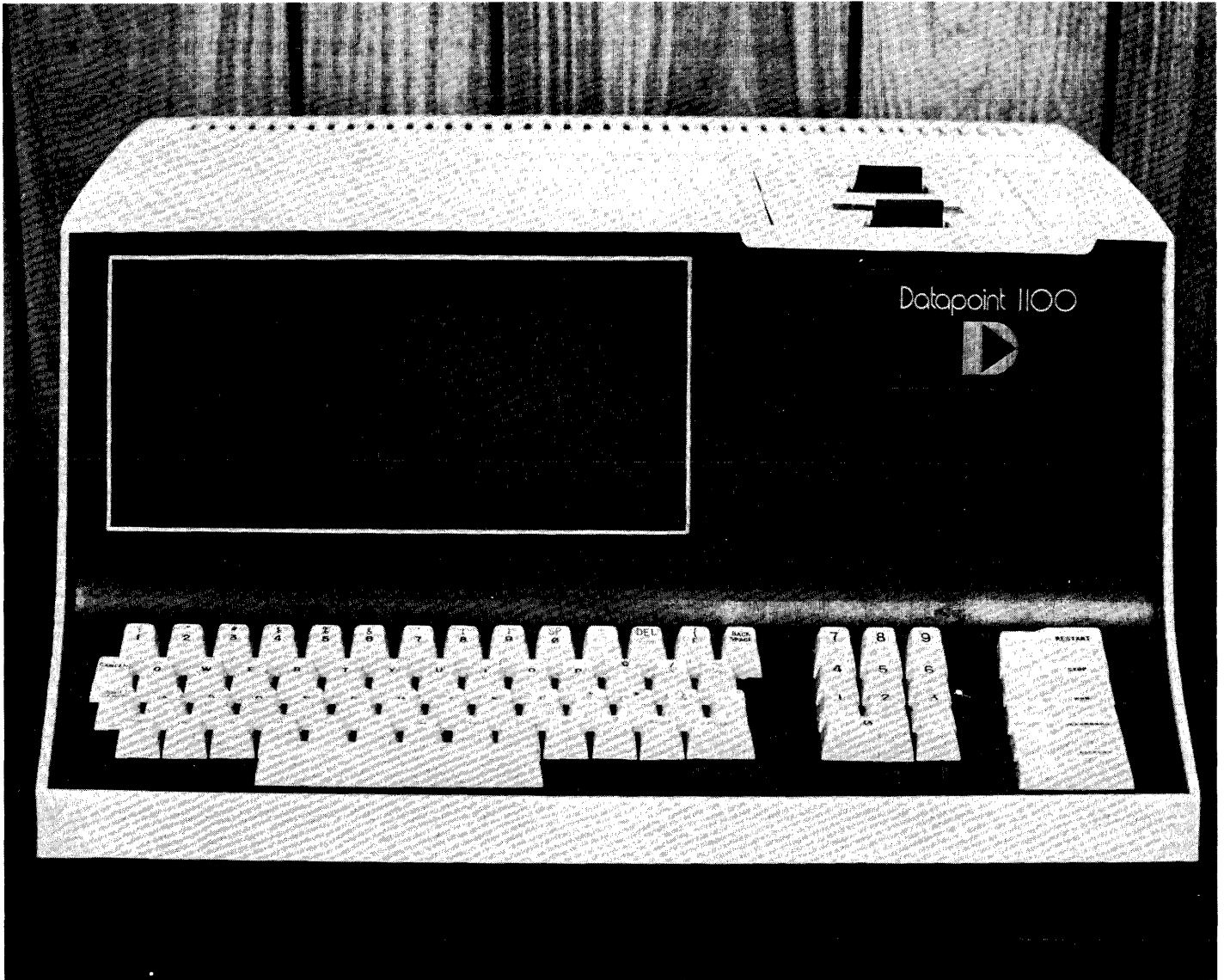
F. Configuring Datashare (DSCON)

G. DOS. Commands

AUTO, MANUAL, AUTOKEY, BACKUP

BLOKEDIT, SORT, SUR

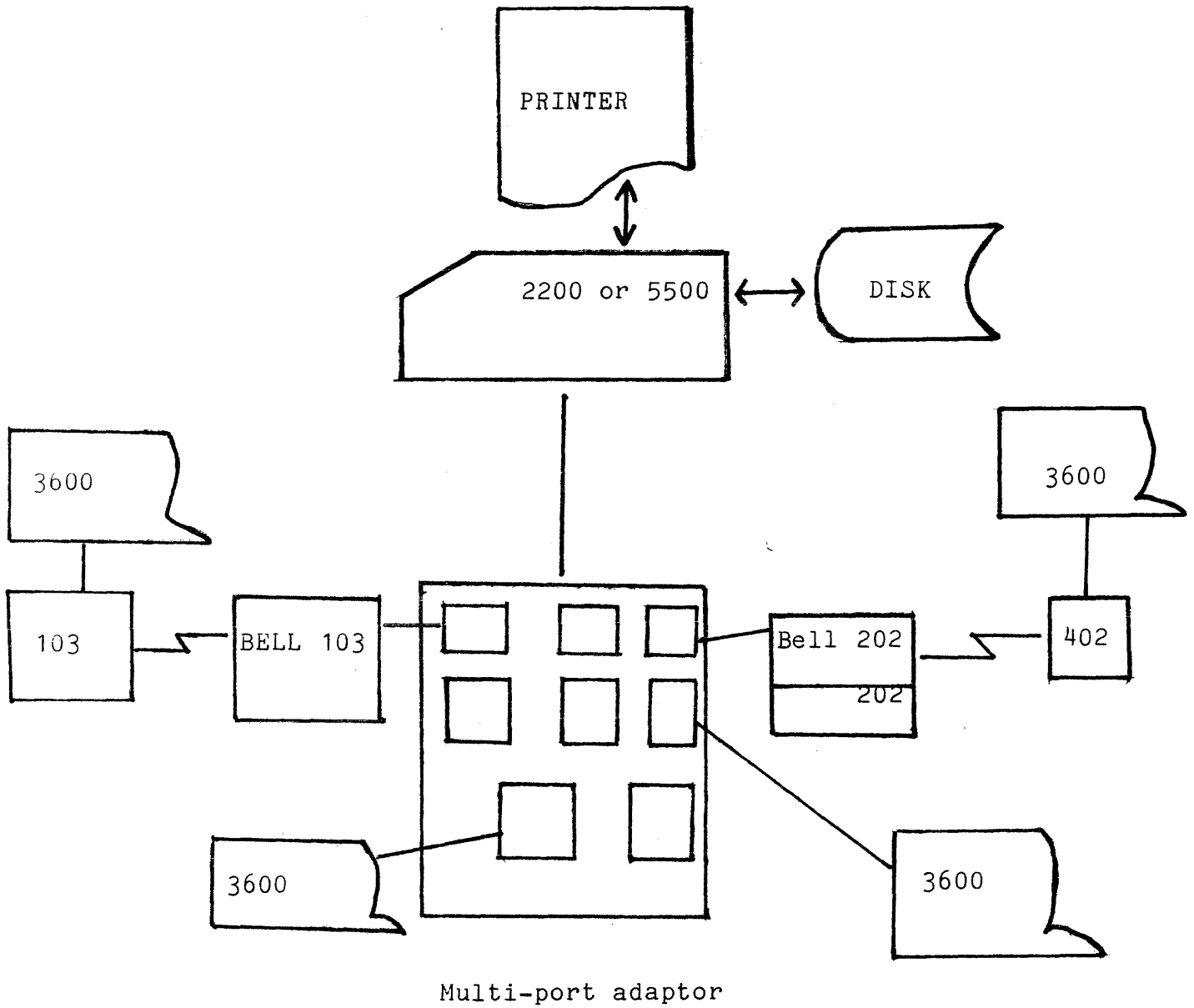
Lab Problems



The Datapoint Processor

Day 1

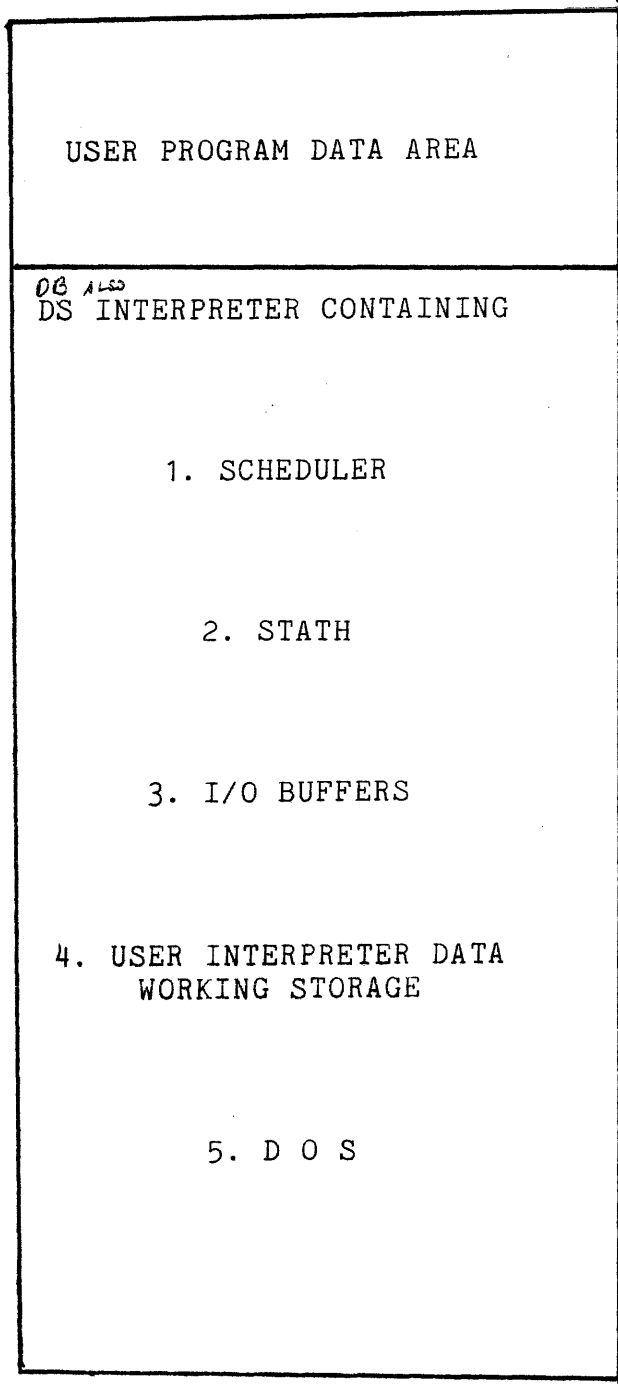
DATASHARE III
HARDWARE CONFIGURATION EXAMPLE



Memory map of the DATASHARE INTERPRETER SYSTEM:

256 BYTE PAGE
CONTROLLER BUFFER
DISKETTE

3K



CONTROLLER PAGE

ARITHMETIC

OP SYS

DISPLAY AND KEYIN INSTRUCTION

NAME	DIM	20
ADDR	DIM	20
CITY	DIM	15
STATE	DIM	2
ZIP	FORM	5
ANS	FORM	1

DISPLAY *ES,"NAME:Ø",*N,"CITY:Ø":
 *N,"STATE:Ø",*P40:4,"ZIP:Ø"

*N,"ADDRESS"

KEYIN *P10:1,NAME,*P10:2,ADDR,*P10:3,CITY:
 *P10:4,STATE,*P45:4,ZIP,*P1:8,"MORE":
 "ØTO DISPLAY - YES = 1, No = 2Ø",ANS

NAME:	
ADDRESS:	
CITY:	
STATE:	ZIP:

DISPLAY

NAME:	John Jones
ADDRESS:	12 Main Street
CITY:	San Antonio
STATE:	Texas
	ZIP: 78229
MORE TO DISPLAY-YES=1,NO=2 ■	

KEYIN

ARITHMETIC INSTRUCTIONS
AND FLAGS

1. Rounding will occur on right truncation if the truncated digit is ≥ 5 .
2. Left truncation occurs if the receiving (result) field is smaller than the sending field (after decimal point alignment).
3. Literals may be used only in the source (sending) field-- never in the result (receiving) field.
4. LESS flag indicates a negative result.
5. EQUAL (ZERO) flag indicates a zero result.
6. OVER flag indicates left truncation ONLY.

Class Problem to Illustrate Truncation

FOR	FORM	"4"
ONE	FORM	"1"
N	FORM	"2"
CTR	FORM	"4"
X	FORM	4.1
PT5	FORM	"0.5"
A1	SUB	ONE,CTR
	GOTO	A5 if Zero
	ADD	PT5,X
	GOTO	A1
A5	ADD	FOR,N
	COMPARE N	TO ONE
	STOP	If Zero [Equal]
	ADD	ONE,N
	GOTO	A1

What is the value of X when the STOP instruction is executed?

DISK

Physical Record Size

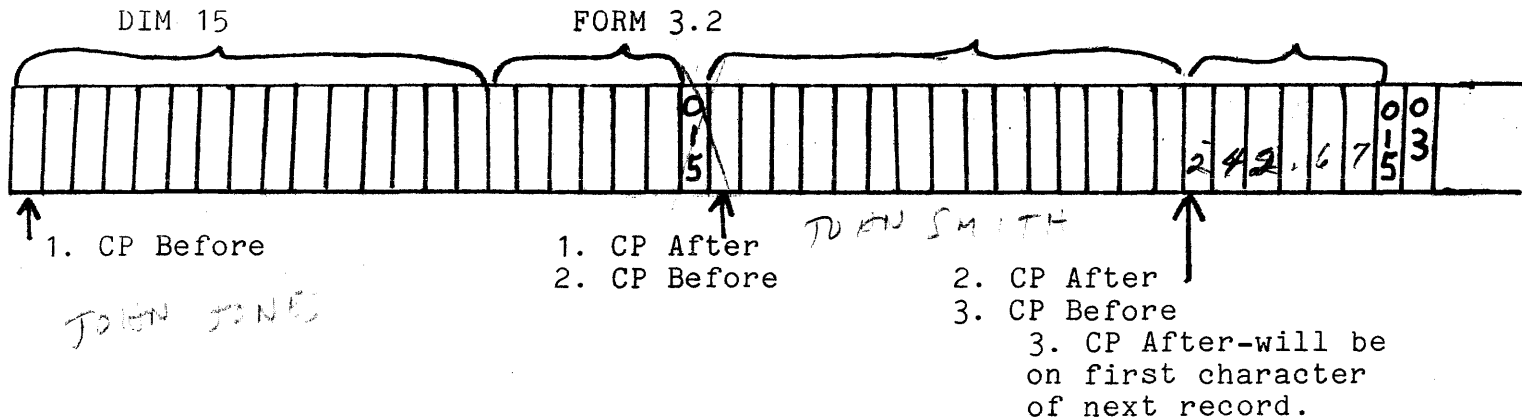
Hardware	256 Bytes
Software	249 Bytes

Datapoint Standards

Ø15	- End of Logical Record
Ø3	- End of Physical Record
Ø11	- Space Compression Character
Ø,Ø,Ø,Ø,Ø,Ø,Ø3	- End of File

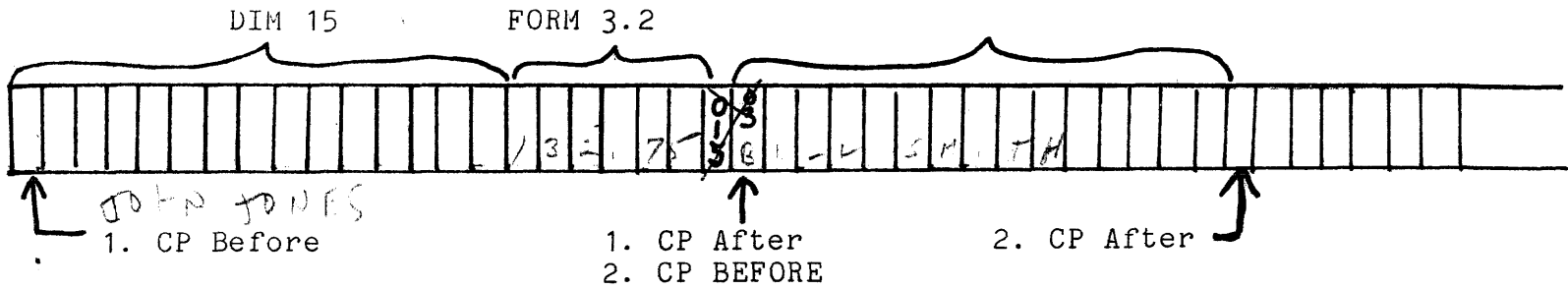
DISK I/O

Sequential Read and the Character Pointer



- JOHN JONES
1. READ FL, SEQ; NAME
 2. READ FL, SEQ; NAME;
 BILL SMITH
 3. READ FL, SEQ; BAL
 242.67

Sequential Writes and the Character Pointer



1. WRITE FL, SEQ~~;~~*-, NAME, BAL; *JOHN JONES*
2. WRITE FL, SEQ~~;~~NAME; *BILL SMITH*

Note: *- in first WRITE turned off space compression

Octal

Any number from 0 thru 7 can be expressed in 3 binary bits. Octal uses only 8 digits (0-7). There is no 8 or 9 in octal.

<u>Decimal</u>	<u>Binary</u>
0	000
1	001
2	010
3	011
4	100
5	101
6	110
7	111

Memory in the Datapoint Processors consists of 8-bit bytes

e.g. 0 1 0 1 1 0 0 1

To express the 8 bits as an octal number, starting from right to left, take groups of 3 bits and convert it to the correct decimal digit (0 thru 7). Since there will be 2 groups of 3 bits, and one group of 2 bits, the 2 high order bits can only range from 00 to 11.

e.g. 01 010 101 = 0125
11 110 111 = 0367

Note that a lead 0 is used by Datapoint Software to indicate that this is an octal value rather than a decimal number.

Decimal to Octal Conversion

1. Decimal Values can be converted to octal by dividing the value by 8, and retaining the remainder. The remainder becomes the right most (least significant) digit of the octal number. Example: $489_{(10)} = ???_{(8)}$

	Remainder	
eg. $8 \overline{) 489}$	1	Decimal--> Octal $489 = ??1_{(8)}$

2. Divide the result of the previous division by 8, again retaining the remainder.

	Remainder	
eg. $8 \overline{) 61}$	5	Decimal--> Octal $489 = ?51_{(8)}$

3. Continue dividing the result of the previous calculation by 8, retaining the remainder, until no more division is possible.

	Remainder	
eg. $8 \overline{) 7}$	7	Decimal--> Octal $489 = 751_{(8)}$

Example: $79_{(10)} = ???_{(8)}$

	Remainder	
$8 \overline{) 1}$	1	
$8 \overline{) 9}$	1	
$8 \overline{) 79}$	7	$= 117_{(8)}$

Octal to Decimal Conversion

Positional notation is one of the simplest methods for converting octal to decimal. Positional notation is multiplying each digit by the base (octal is base 8, decimal is base 10) Raised to a certain power relative to it's position.

$$\begin{array}{rcll} \text{eg. decimal} & 5498 & = & 5000 = 5(1000) = 5(10^3) \\ & & + & 400 = 4(100) = 4(10^2) \\ & & + & 90 = 9(10) = 9(10^1) \\ & & + & 8 = 8(1) = 8(10^0) \end{array}$$

Therefore, octal ---> decimal

$$\begin{array}{rcll} \text{Octal} & 751 & = & 7(8^2) = 7(64) = 448 \\ & & + & 5(8^1) = 5(8) = 40 \\ & & + & 1(8^0) = 1(1) = 1 \\ & & & 489_{(10)} \end{array}$$

$$\begin{array}{rcll} \text{Octal} & 117 & = & 1(8^2) = 1(64) = 64 \\ & & & 1(8^1) = 1(8) = 8 \\ & & & 7(8^0) = 7(1) = 7 \\ & & & 79_{(10)} \end{array}$$

Day 2

DISK FILES

BUFFERS: Each Separate

Each 256 Bytes

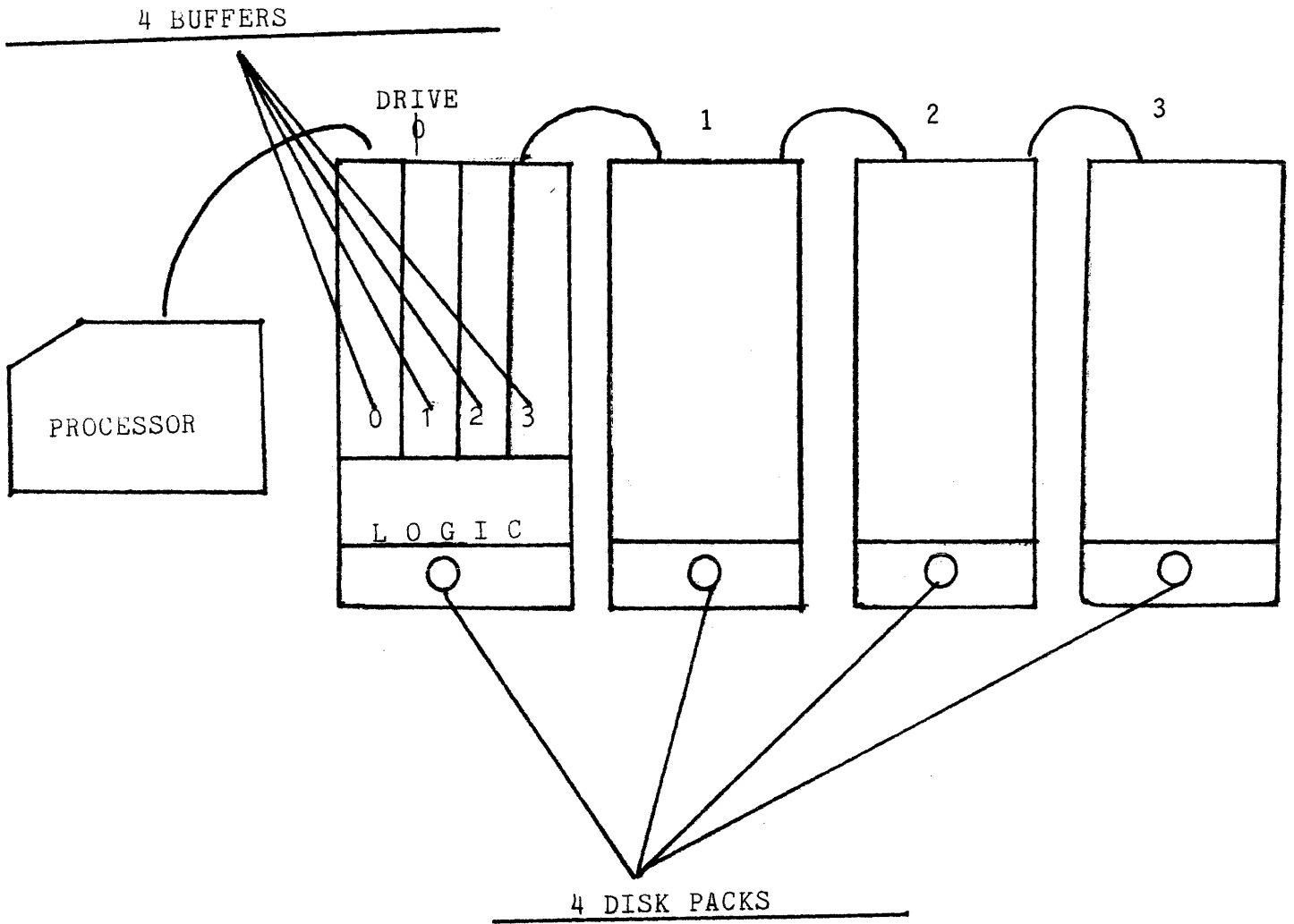
All I/O goes thru controller buffer in a two step operation.

PROCESSOR -----> BUFFER -----> DISK

The buffers are not assigned to any particular drive.

This is controlled by the software.

DISK FILE EXAMPLE



CARTRIDGE DISK

A BYTE equals 8 Bits

A SECTOR equals 256 Bytes

A CYLINDER equals 48 Sectors

A CLUSTER equals 6 Sectors

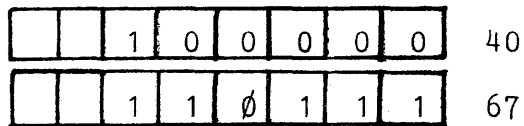
A TRACK has 24 Sectors

Track 0 is on top side of disk.

Track 1 is on bottom side of disk.

Track 0 has sectors 0 - 27.

Track 1 has sectors 40 - 67.



A DISK has 203 Cylinders per disk.

Cluster 0 - 3 are on the top side.

Cluster 4 - 7 are on the bottom side.

On Disk number 0 there is only 2.2 million available Bytes because of having the disk operating system on cylinders 1 - 4, and the DOS tables on cylinder 0.

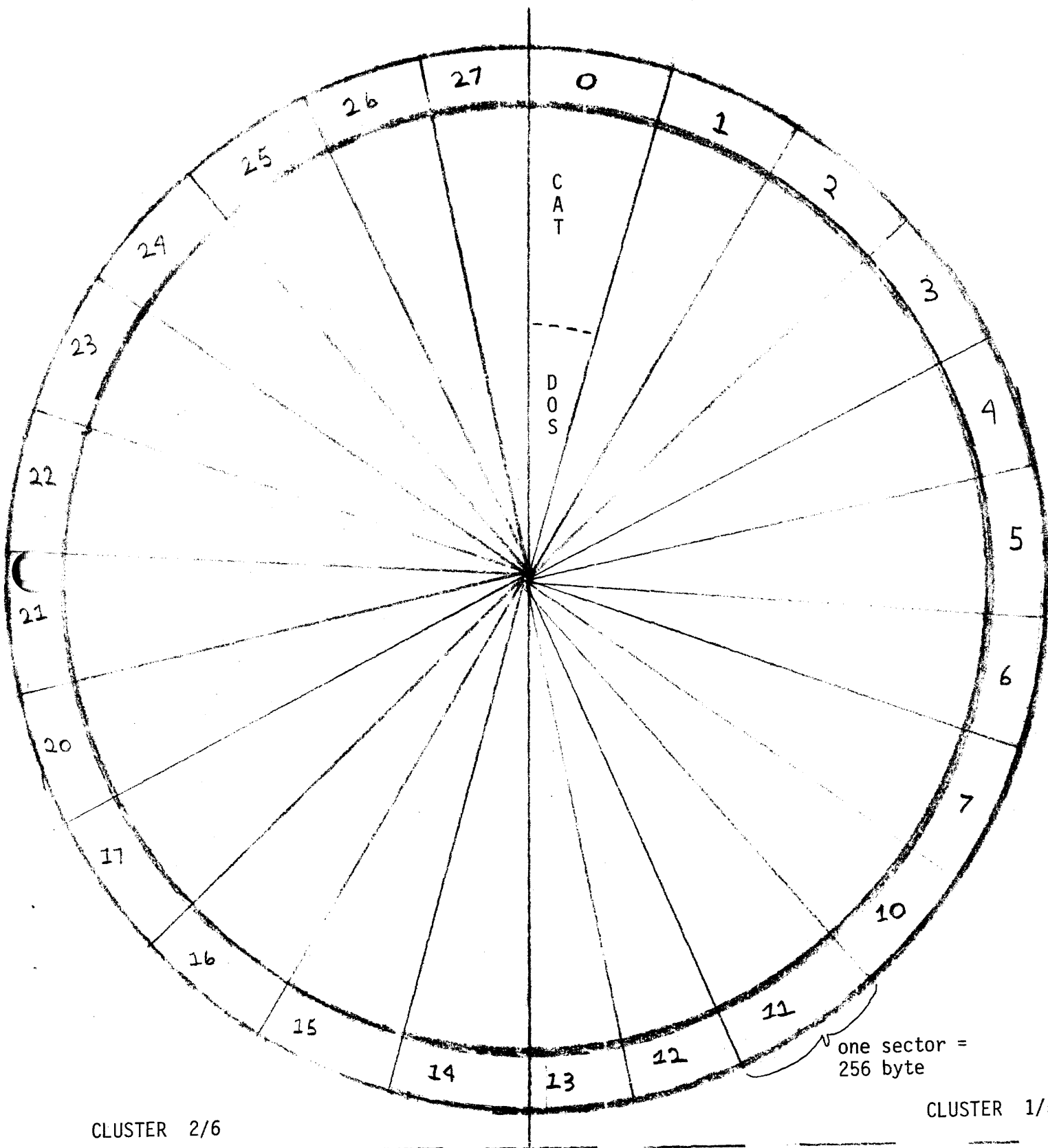
The maximum number of files on any given disk is 256.

DISK

CLUSTER 3/7

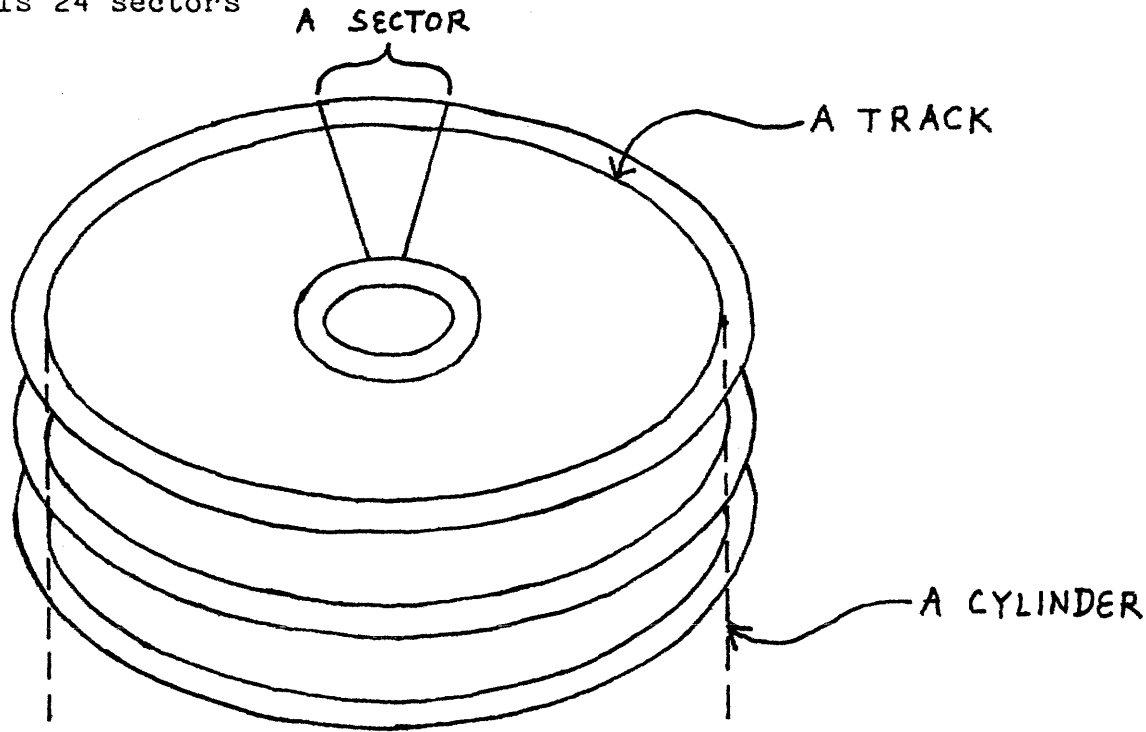
ONE CYLINDER

CLUSTER 0/4



MASS STORAGE DISK

A Sector equals 256 bytes
A Cylinder equals 192 sectors
A Track equals 24 sectors



A disk has 203 Cylinders per surface

A Cluster = 1 cylinder on 1 Track

FLEXIBLE DISK

Physical Characteristics

- 1 Sector = 128 Bytes
- 1 Diskette = 256,256 Bytes
- 1 Inch = 3200 bits (Inside track)
- 1 Track = 26 Sectors
- 1 Diskette = 77 Tracks

Logical I/O

Sectors are paired in controller

13 Sectors / track

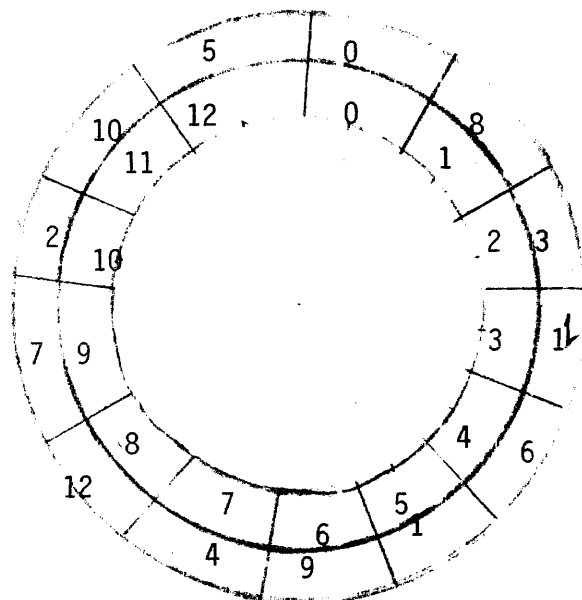
256 Bytes / Sectors

Only 12 sectors used for Data

13th Sector used for Directory

Track 0 reserved for IBM compatibility

The logical numbering of the sectors of each track is staggered to take advantage of rotational delay. Thus, three sequential logical sectors may be read per revolution of the disk.

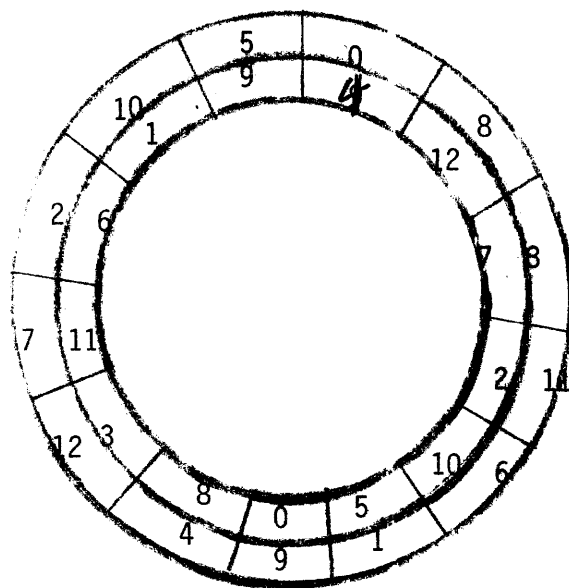


Logical Sector Numbering

Physical Sector Numbering

Logical Sector Number 12 is always the directory for each track.

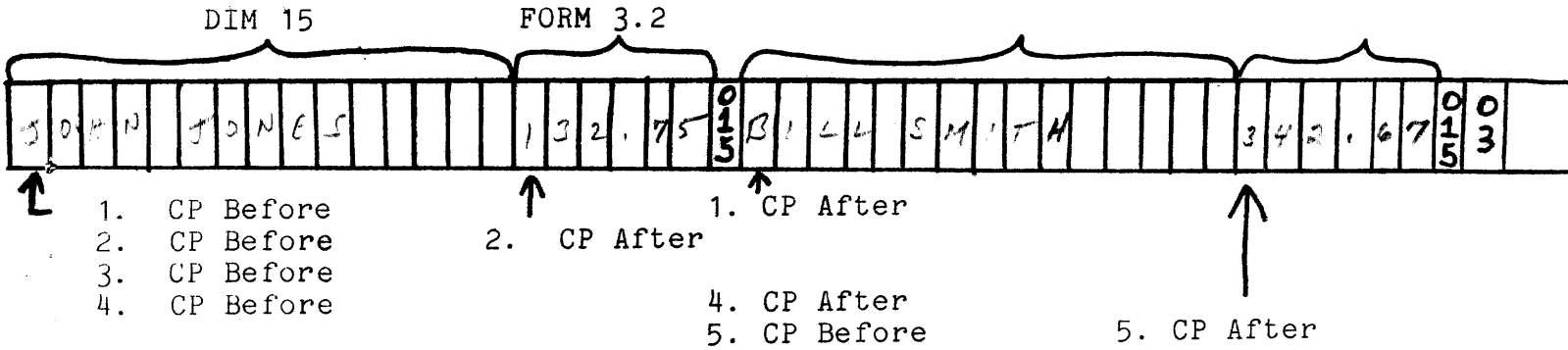
The relationship of logical sectors from track to track is staggered to take advantage of rotational delay. Thus, three sequential directory segments may be read in a single revolution of the disk.



Logical Sectors are staggered from track to track.

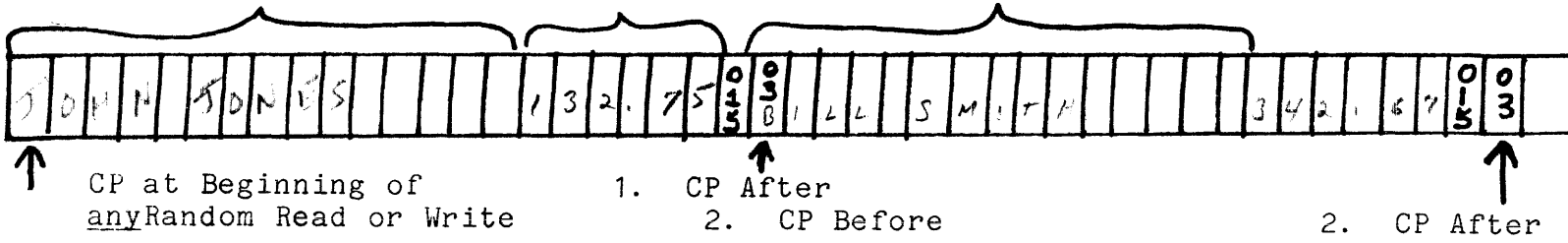
Numbering is re-aligned to physical sector 0 every fourth track.

Random Reads and the Character Points



1. READ FL,RN;NAME
 2. READ FL,RN;NAME;
 3. READ FL,RN;BAL; ----> this will result in a Format Error
 4. READ FL,RN;NAME
 5. READ FL,SEQ;NAME;
- } a Random Read followed by a Sequential Read allows processing of blocked records

Random Writes and the Character Pointer



Sequential Write
to JOHN JONES

1. WRITE FL,RN;NAME,BAL
2. WRITE FL,SEQ;NAME,BAL



a Random Write followed by a Sequential Write allows blocking

DOS COMMAND HANDLER

DISK FILES

NAME/EXT - 1 - 3 characters

EXT TABLE

CMD- Written in assembler, executable directly, system Utilities.

ABS- Assembler program, user program

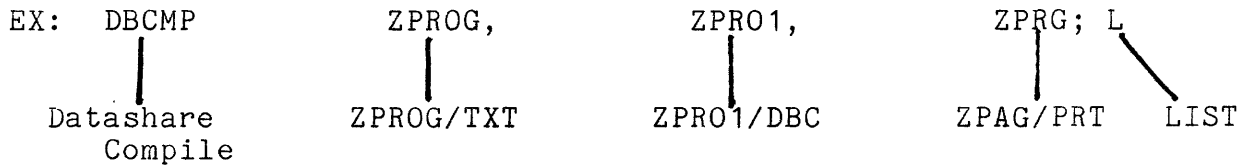
TXT- Text, source and data

DBC- Databus/Datashare, user programs

OVn - Overlays for systems programs

SYS- System files and programs that are unusual.

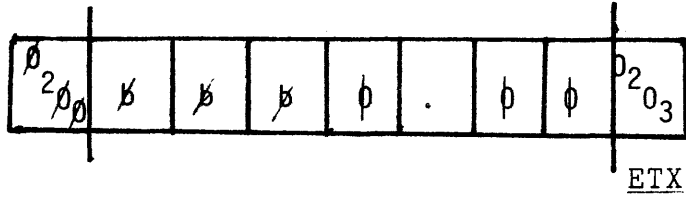
PRT- Print files



VARIABLES IN MEMORY

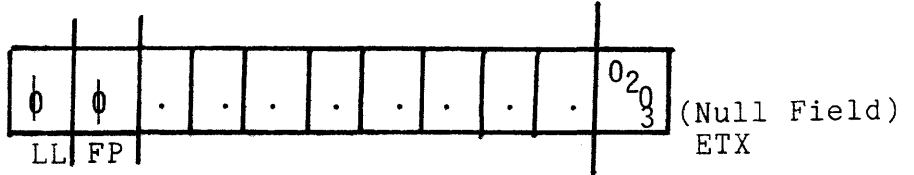
numeric

NUM FORM 4.2



String

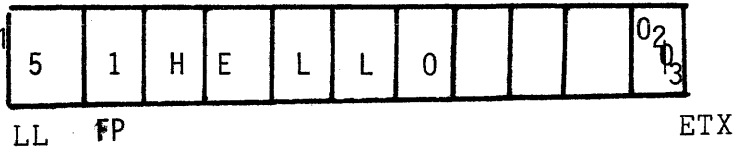
ALF1 DIM 8



e.g. After

KEYIN

ALF1



TO GET LEADING ZEROES

NUM FORM 4.2

SNUM DIM 7

0		E
2		0
0	0 35.27	S
0		

NUM AFTER KEYIN

		0 35.27	E
7	1		0
			S

MOVE NUM TO SNUM

L F
L P

```
L1  CMATCH "0", SNUM
      GOTO FINISH IF NOT EQUAL
      CMOVE "0", SNUM
      BUMP  SNUM
      GOTO L1
```


<u>VAR</u>	<u>LL</u>	<u>FP</u>	<u>CONTENTS</u>	
STRING1	4	2	ABCDXLM	ETX
STRING2	6	3	DOGCAT	ETX
MOVE STRING1 TO STRING2				
STRING2	3	1	BCDCAT	ETX
STRING2	6	3	DOGCAT	ETX
MOVE "HELLO" TO STRING2				
STRING2	5	1	HELLOT	ETX
STRING1	9	3	AB100.327	ETX
NUMBER	0200		Ø39.00	ETX
MOVE STRING1 TO NUMBER				
NUMBER	0200		100.33	ETX

MATCH

<u>SOURCE</u>	<u>DESTINATION</u>	<u>FLAG set</u>
ABCDE	ABCD	EQUAL (Chars Compared are Equal) LESS (Dest. String is Shorter)
ABC	Z	NOT EQUAL (Chars Compared are not equal) NOT LESS (ASCII Value of Dest. String Not Less Than Source)
ZZZ	AAA	NOT EQUAL LESS (Value of Dest. Less Than Source)
ABC	ABC	EQUAL NOT LESS (Dest. String Length Not Less Than Source String Length)
ABCD	ABCDE	EQUAL (# of Chars Compared are Equal) NOT LESS (Dest. String Length Not Less Than Source String Length)

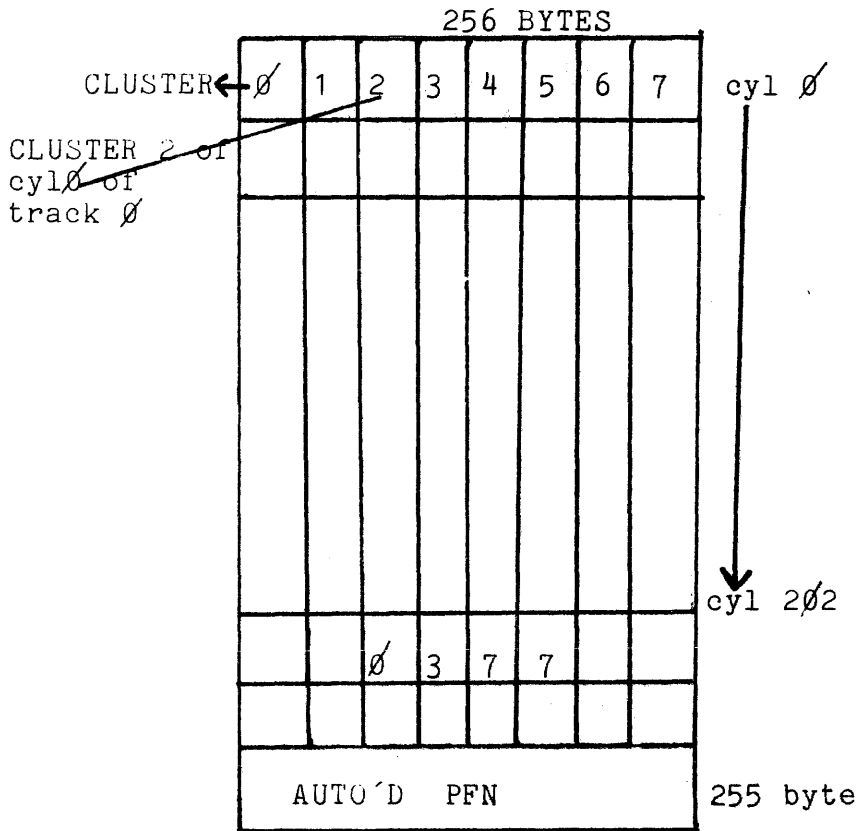
KEYBOARD CODING (ASCII)

A-101	a-141	0-060	:	-072
B-102	b-142	1-061	;	-073
C-103	c-143	2-062	<	-074
D-104	d-144	3-063	=	-075
E-105	e-145	4-064	>	-076
F-106	f-146	5-065	?	-077
G-107	g-147	6-066	!	-133
H-110	h-150	7-067	\	-176
I-111	i-151	8-070]	-135
J-112	j-152	9-071	^	-136
		Space -040	-	-137
K-113	k-153			
L-114	l-154	!-041	@	-100
M-115	m-155	"-042	{	-173
N-116	n-156	#-043	\	-134
O-117	o-157	\$-044	'	-140
P-120	p-160	--045	!	-174
Q-121	q-161	&-046	}	-175
R-122	r-162	^-047	Enter	-015
S-123	s-163	(-050	Cancel	-030
T-124	t-164)-051	Backspace	-010
U-125	u-165	*-052	Del	-177
V-126	v-166	+--053		
W-127	w-167	, -054		
X-130	x-170	--055		
Y-131	y-171	. -056		
Z-132	z-172	/-057		

Day 4

Up to 256 FILES does

File Structure



CLUSTER ALLOCATION TABLE

Will indicate if cluster available for use or in use

1 bit for each cluster

Each bit points to a cluster

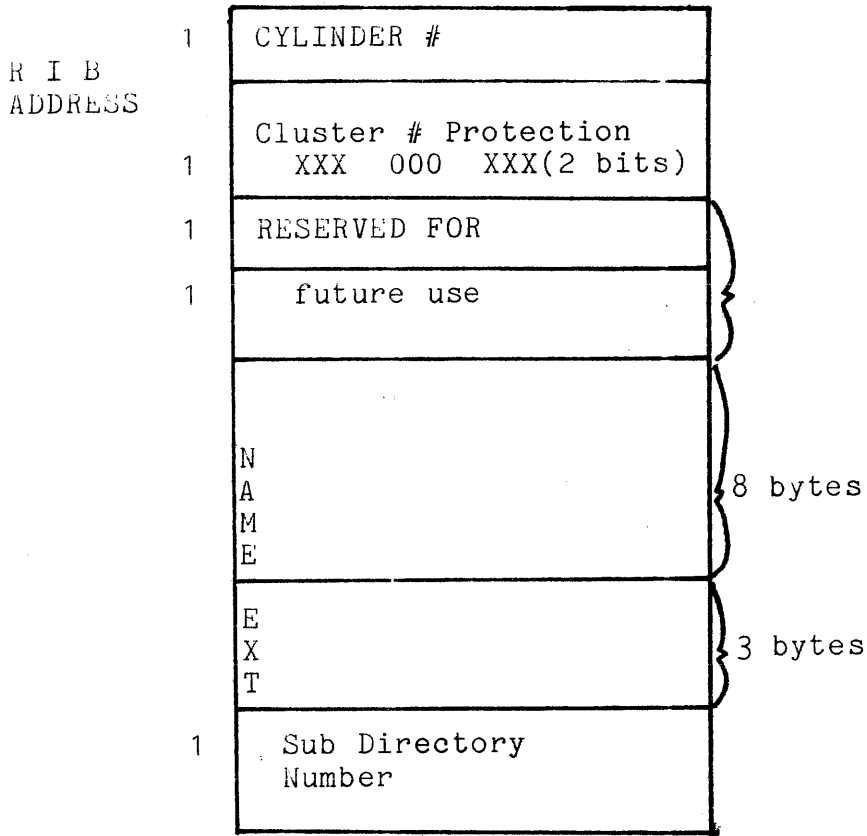
Each byte points to a cylinder

1 = cluster occupied

AUTO'D PFN is 0 if not specified otherwise - this is the DOS system program.

DIRECTORY ENTRY FORMAT

A FILE ENTRY



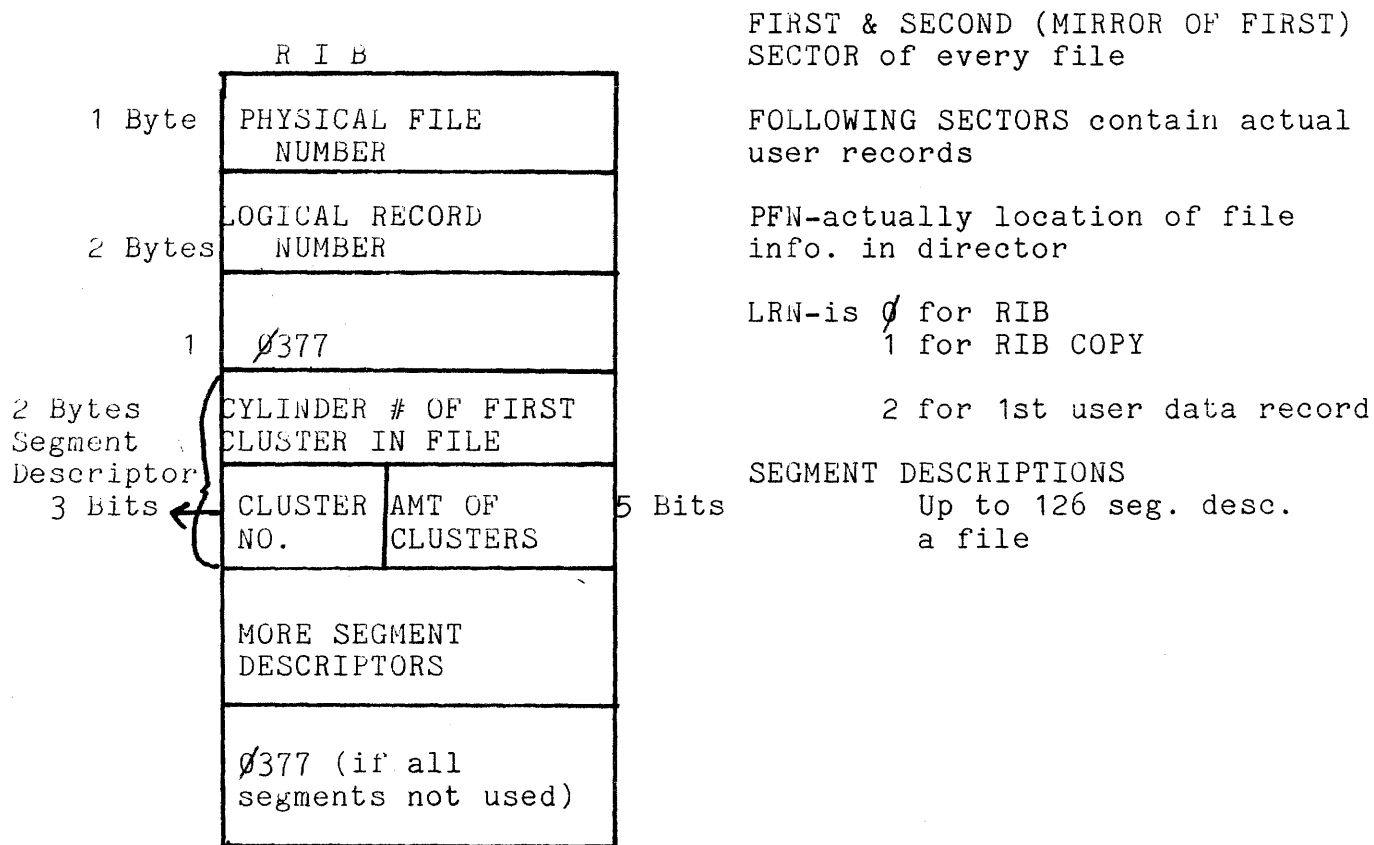
TAKES 16 SECTORS
RIB-Retrieval information
block where location
of data records is described
and actual data starts

1 entry = 16 BYTES

WRITE PROTECTION gives
both write and delete

- 16 SECTORS
- 16 ENTRIES/SECTOR-each entry has its own physical file number
(physical location in directory)
- 16 BYTES PER ENTRY

RETRIEVAL INFORMATION BLOCK



Knowing cyl # and cluster # is sufficient because all segments must start at the beginning of a cluster and the cluster number is actually giving track and sector.

- 3 ways to run out of space:
- no clusters left in C.A.T
 - no room in directory
 - no more segment descriptors in R.I.B.

NAMEFILE

ISAM

Aaron
Bob
Cal
Fred
Ava
Ira
Adam
Bill
Shirley
Gail
Mona
Jill
Neil
Ed
Lou
Dave
Steve
Carl
Betty
Ralph
Alan
Lynn
Mary
Tom
Jack
George
Ron
Sally
Larry
Sandy
Bobby
Kevin

AARON
JACK

AARON
CAL

JACK
NEIL

Aaron
Adam
Alan
Ava
Betty
Bill
Bob
Bobby

Cal
Carl
Dave
Ed
Fred
Gail
George
Ira

Jack
Jill
Kevin
Larry
Lou
Lynn
Mary
Mona

Neil
Ralph
Ron
Sally
Sandy
Shirley
Steve
Tom

Index Sequential Reads

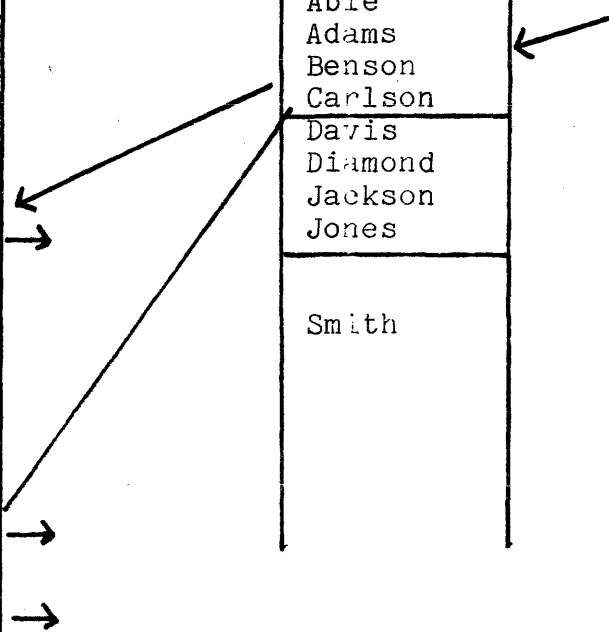
/TXT File

/ISI File

Jones
Adams
Smith
Benson
Able
Diamond
Carlson
Jackson
Davis

Able
Adams
Benson
Carlson
Davis
Diamond
Jackson
Jones
Smith

- ① READ IFL, KEY;NAME,etc.
(if KEY = Benson)
- ② READKS IFL;NAME,etc.
- ③ READS IFL,SEQ;NAME,etc.



ISAM FORMULA

Lowest Level:

$$\# \text{ of records/sectors} = \frac{250}{\text{Key Length} + 7} \quad \begin{array}{l} \text{Round} \\ \text{Down} \end{array}$$

$$\# \text{ of sectors} = \frac{\# \text{ of Records}}{\# \text{ of Records/Sectors}} \quad \begin{array}{l} \text{Round} \\ \text{Up} \end{array}$$

Next Level:

$$\# \text{ of records/sectors} = \frac{250}{\text{Key Length} + 3}$$

$$\# \text{ of sectors} = \frac{\# \text{ of Sectors on Previous Level}}{\# \text{ of Records/Sector}}$$

Next Level:

Repeat next level formula above by using the previous # of sectors.

ISAM FORMULA

EXAMPLE: Needed Information

File Length = 9500 Records
Key Length = 15 Characters

LOWEST LEVEL:

$$\# \text{ of Records/Sector} = \frac{250}{22} = 11.36 = 11$$

$$\# \text{ of Sectors} = \frac{9500}{11} = 863.63 = 864$$

NEXT LEVEL:

$$\# \text{ of Records/Sector} = \frac{250}{18} = 13.89 = 13$$

$$\# \text{ of Sectors} = \frac{864}{13} = 66.46 = 67$$

NEXT LEVEL:

$$\# \text{ of Records/Sector} = \frac{250}{18} = 13.89 = 13$$

$$\# \text{ of Sectors} = \frac{67}{13} = 5.15 = 6$$

ISAM FORMULA

EXAMPLE:

NEXT LEVEL:

$$\# \text{ of Records/Sector} = \frac{250}{18} = 13.89 = 13$$

$$\# \text{ of Sectors} = \frac{6}{13} = 1$$

Add Header Record = 4

ISAM EXAMPLE TALE

Lowest Level	864
Next Level	67
Next Level	6
Next Level	1
Header Record	4
Total	<u>942</u>

ISAM EXAMPLE #2

Needed Information

File Length = 9500 Records

Key Length = 5 Characters

ISAM EXAMPLE # 2 TABLE

Lowest Level	475
Next Level	16
Next Level	1
Header Record	4
TOTAL	<u>496</u>

Day 5

NOTES

Sample Solutions

to

Lab Problems

Lab Problem
Datashare

Sequential File Processing

There exists a file on disk called NAMEFILE with the following layout:

ACCT	FORM	4 (Account Number)
NAME	DIM	20
STREET	DIM	35
CITY	DIM	20
STATE	DIM	15
ZIP	DIM	5
BAL	FORM	4.2 (Current Balance)

NAMEFILE is to be read sequentially and a new file is to be created with update information.

The information in NAMEFILE is to be displayed on the screen as each record is read.

For each account, allow for a series of deposits and withdrawals against the current balance to be keyed in by the operator. Update the balance, display the new balance (do not allow an account to be overdrawn), and write the updated record to the newly created file.

NOTE: The first data record on Namefile contains the number of data records on the file. For the purpose of this lab problem, it should be by-passed.

```

55.      .
56.      . *** WITHDRAWAL ROUTINE ***
57.      .
58. 01570 WITHRTN KEYIN *P1:11,*EF,"WITHDRAWAL AMOUNT: ",WITHD
59. 01623         SUB      WITHD,HBAL
60. 01630         GOTO     OVERDS IF LESS
61. 01634         MOVE     HBAL,BAL
62. 01641         GOTO     TOOBIGS IF OVER
63. 01645         DISPLAY *P1:9,*EL,"NEW BALANCE: ",HBAL
64. 01672         GOTO     NEXT
65.      .
66.      . *** ERROR ROUTINE FOR OVERDRAWN ACCOUNTS ***
67.      .
68. 01675 OVERDA  SUB      DEP,HBAL
69. 01702         GOTO     OVERD
70. 01705 OVERDS  ADD      WITHD,HBAL
71. 01712 OVERD   DISPLAY *P10:12,*EL,"*****ACCOUNT OVERDRAWN - LAST TRANS":
72. 01763         "ACTION NOT ACCEPTED*****",*W,*W,*W
73. 02026         BEEP
74. 02027         DISPLAY *P1:9,*EL,"CURRENT BALANCE: ",HBAL
75. 02060         GOTO     NEXT
76.      .
77.      . *** ERROR ROUTINE IF BALANCE TOO LARGE FOR FIELD ***
78.      .
79. 02063 TOOBIGA SUB      DEP,HBAL
80. 02070         MOVE     HBAL,BAL
81. 02075         GOTO     TOOBIG
82. 02100 TOOBIGS ADD      WITHD,HBAL
83. 02105         MOVE     HBAL,BAL
84. 02112 TOOBIG  DISPLAY *P10:12,*EL,"*****BALANCE EXCEEDS LIMITS - LAST TRANS":
85. 02170         "ACTION NOT ACCEPTED*****",*W,*W,*W
86. 02225         BEEP
87. 02226         DISPLAY *P1:9,*EL,"CURRENT BALANCE: ",BAL
88. 02257         GOTO     NEXT
89.      .
90.      . **** WRITE NEWLY UPDATED RECORD ****
91.      .
92. 02262 WRITE   WRITE   OUTFL,SEQ;ACCT,NAME,STREET,CITY,STATE,ZIP,BAL
93. 02306         GOTO     START
94.      .
95.      . *** NO MCRE PROCESSING TO BE DONE ON THIS FILE ***
96.      .
97. 02311 FINISH  READ     INFL,SEQ;ACCT,NAME,STREET,CITY,STATE,ZIP,BAL
98. 02335         GOTO     EOJ IF OVER
99. 02341 DONE    WRITE   OUTFL,SEQ;ACCT,NAME,STREET,CITY,STATE,ZIP,BAL
100. 02365         GOTO     FINISH
101.      .
102.      . ***** END OF JOB *****
103.      .
104. 02370 EOJ     WEOF     OUTFL,SEQ
105. 02404         CLOSE    INFL
106. 02407         CLOSE    OUTFL
107. 02412         DISPLAY *ES,*P10:5,"*****END OF PROGRAM*****",*W,*W,*W,*W
108. 02456         STOP

```

```

1.      . ***** SEQUENTIAL FILE PROCESSING LAB - DAY 1 *****
2.      .
3.      00000 ACCT    FORM  4 ✓
4.      00006 NAME    DIM  20
5.      00035 STREET  DIM  35
6.      00103 CITY    DIM  20
7.      00132 STATE  DIM  15
8.      00154 ZIP     DIM   5
9.      00164 BAL     FORM  4.2
10.     00175 HBAL    FORM  5.2
11.     00207 WITHD   FORM  4.2
12.     00220 DEP     FORM  4.2
13.     00231 INFL    FILE
14.     00252 OUTFL   FILE
15.     00273 SEQ     FORM  "-1"
16.     00277 ANS     FORM   1
17.     .
18.     . *** BEGINNING OF PROGRAM ***
19.     .
20.     01001 BEGIN   OPEN    INFL,"NAMEFILE"
21.     01024         PREPARE OUTFL,"ZNEW"
22.     01043         READ    INFL,SEQ;ACCT          (BYFASSES FIRST RECORD)
23.     .
24.     . ***** MAIN PROCESSING *****
25.     .
26.     01053 START   READ    INFL,SEQ;ACCT,NAME,STREET,CITY,STATE,ZIP,BAL
27.     01077         GOTO    EOJ IF OVER
28.     01103         DISPLAY *ES,"NAME:      ",NAME,*N,"STREET:  ",STREET,*N:
29.     01135         "CITY:      ",CITY,*N,"STATE:   ",STATE,*P45:4:
30.     01167         "ZIP:      ",ZIP,*P1:6,"ACCOUNT: ",ACCT,*P1:8,"ORIG":
31.     01223         "INAL BALANCE: ",BAL
32.     01244         MOVE    BAL,HBAL
33.     01251 NEXT   KEYIN   *P1:11,*EF,"KEY 1=DEPOSIT, 2=WITHDRAWAL, 3=NEXT ":
34.     01322         "RECORD, 4=STOP PROCESSING: ",ANS
35.     01360         COMPARE "1",ANS
36.     01401         GOTO    DEPRTN IF EQUAL
37.     01405         COMPARE "2",ANS
38.     01420         GOTO    WITHRTN IF EQUAL
39.     01424         COMPARE "3",ANS
40.     01437         GOTO    WRITE IF EQUAL
41.     01443         COMPARE "4",ANS
42.     01456         GOTO    DONE IF EQUAL
43.     01462         BEEP
44.     01463         GOTO    NEXT
45.     .
46.     . *** DEPOSIT ROUTINE ***
47.     .
48.     01466 DEPRTN KEYIN   *P1:11,*EF,"DEPOSIT AMOUNT: ",DEP
49.     01516         ADD     DEP,HBAL
50.     01523         GOTO    OVERDA IF LESS
51.     01527         MOVE    HBAL,BAL
52.     01534         GOTO    TOOBIGA IF OVER
53.     01540         DISPLAY *P1:9,*EL,"NEW BALANCE: ",HBAL
54.     01565         GOTO    NEXT

```


PAGE 2

TODD3/TXT:DR0

```
55.      .
56.      . *** ERROR ROUTINE FOR INVALID ACCOUNT NUMBER ***
57.      .
58.      01545 NOREC   DISPLAY  *P1:12,*EL,"INVALID ACCOUNT NUMBER",*W,*W
59.      01603      BEEP
60.      01604      GOTO    GO
61.      01607      STOP
62.      01610      STOP
```

```

1.          . ***** RANDOM FILE PROCESSING LAB - DAY 2 *****
2.          .
3.          00000  ANS      FORM  1
4.          00003  ACCT     FORM  4
5.          00011  NAME     DIM   20
6.          00040  STREET   DIM   35
7.          00106  CITY     DIM   20
8.          00135  STATE    DIM   15
9.          00157  ZIP      DIM    5
10.         00167  BAL      FORM  4.2
11.         00200  RN       FORM  4
12.         00206  SEQ      FORM  "-1"
13.         00212  INFL     FILE
14.         00233  AFL      FILE
15.         00254  LOWNUM   FORM  2
16.         00260  RECNUM   FORM  4
17.         00266  LOW      FORM  2
18.          .
19.          . *** BEGINNING OF PROGRAM ***
20.          .
21.         01001  START    OPEN      INFL,"NAMEFILE"
22.         01024          OPEN      AFL,"ACCTFILE"
23.         01047          DISPLAY   *ES,"PROGRAM START-UP",*W,*W,*W,*W
24.         01076  GO      KEYIN     *ES,"KEYIN ACCOUNT NUMBER: ",RN
25.         01131          MOVE      RN,LOW
26.         01136          DIV       "100",RN
27.         01153          MULT     "2",RN
28.          .
29.          . *** FIND RECORD NUMBER ON ACCOUNT FILE ***
30.          .
31.         01166  AREAD1  READ      AFL,RN;LOWNUM,RECNUM;
32.         01200  COMPA   COMPARE   "0",RECNUM
33.         01213          GOTO     NOREC IF EQUAL
34.         01217          COMPARE  LOWNUM,LOW
35.         01224          GOTO     OKA IF EQUAL
36.         01230          READ     AFL,SEQ;LOWNUM,RECNUM;
37.         01242          GOTO     COMPA
38.          .
39.          . *** FIND RECORD ON NAMEFILE ***
40.          .
41.         01245  OKA     READ      INFL,RECNUM;ACCT,NAME,STREET,CITY,STATE,ZIP,BAL
42.         01271          DISPLAY  "ACCOUNT: ",ACCT,*N,"NAME:   ",NAME,*N:
43.         01322          "STREET: ",STREET,*N,"CITY:   ",CITY,*N:
44.         01351          "STATE:  ",STATE,*N,"ZIP:    ",ZIP,*N:
45.         01406          "BALANCE: ",BAL
46.         01422  ASK    KEYIN     *P1:12,*EL,"MORE? 1=YES 2=NO: ",ANS
47.         01454          COMPARE  "1",ANS
48.         01467          GOTO     GO IF EQUAL
49.         01473          COMPARE  "2",ANS
50.         01506          GOTO     ASK IF NOT EQUAL
51.         01512          DISPLAY  *ES,"END OF PROGRAM",*W,*W,*W
52.         01536          CLOSE    INFL
53.         01541          CLOSE    AFL
54.         01544          STOP

```

STRING PROBLEM

STRING DIM 10
COLLAT INIT "ABCDE----->ZØ12----->9"
RESULT DIM 10

1. Assume string contains "ZMNPAB0194"
2. MOVE THE STRING characters to the RESULT field in the order given in COLLAT

EX - STRING ZMNP
 COLLAT ABC----->ZØ12----->9
 RESULT WOULD BE MNPZ

Show string instructions used to accomplish this.

```
      CLEAR     RESULT  
      RESET     COLLAT  
  
P1     RESET     STRING  
P2     MATCH     STRING, COLLAT  
  
      GOTO     P3 IF NOT EQUAL  
      EXTEND     RESULT  
      CMOVE     STRING, RESULT  
  
P3     BUMP     STRING  
      GOTO     P2 IF NO EOS  
      BUMP     COLLAT  
      GOTO     P1 IF NOT EOS  
  
      STOP
```

```

1.      . ***** STRING MANIPULATION LAB - DAY 3 *****
2.      .
3.      00000 MCN      FORM  2
4.      00004 MON1    LIM    2
5.      00011 AMON    DIM    3
6.      00017 CTR     FORM  2
7.      00023 TABLE  INIT   "JANFEBMARAPRMAYJUNJULAUGSEPNOVDEC"
8.      00072 ADATE   INIT   "      , 19  "
9.      00111 NDATE   DIM    12
10.     00130 INDATE  DIM    12
11.     00147 ANS     DIM    1
12.     .
13.     . ***** BEGINNING OF PROGRAM *****
14.     .
15.     01001 START   MOVE    "1",CTR
16.     01014         RESET   TABLE
17.     01020         KEYIN   *ES,"DATE IN ALPHA (MMM DD, YYYY) OR NUMERIC (MMDDYY) ":
18.     01103         INDATE
19.     01106         TYPE    INDATE
20.     01111         GOTO    NUMERIC IF EQUAL
21.     .
22.     . *** ALPHA TO NUMERIC CONVERSION ***
23.     .
24.     01115 ALPHA   MOVE    INDATE,AMON
25.     01122         MATCH   AMON,TABLE
26.     01127         GOTO    OKA IF EQUAL
27.     01133         BUMP    TABLL,3
28.     01137         GOTO    ERROR IF EOS
29.     01143         ADD     "1",CTR
30.     01156         GOTO    ALPHA
31.     01161 OKA     MOVE    CTR,NDATE
32.     01166         RESET   INDATE,6
33.     01172         LENSET  INDATE
34.     01175         RESET   INDATE,5
35.     01201         ENDSET  NDATE
36.     01204         APPEND  INDATE,NDATE
37.     01211         RESET   INDATE,12
38.     01215         RESET   INDATE,11
39.     01221         APPEND  INDATE,NDATE
40.     01226         RESET   NDATE
41.     01232         DISPLAY "YOUR CONVERTED ALPHA DATE IN NUMERIC IS ",NDATE,*W,*W,*W
42.     01311 ASK    KEYIN   "KEYIN TO CONTINUE Y=YES N=NO ",ANS
43.     01352         MATCH   "Y",ANS
44.     01366         GOTO    START IF EQUAL
45.     01372         MATCH   "N",ANS
46.     01414         GOTO    OUT IF EQUAL
47.     01420         BEEP
48.     01421         GOTO    ASK
49.     .
50.     . *** NUMERIC TO ALPHA CONVERSION ***
51.     .
52.     01424 NUMERIC MOVE    INDATE,MON1
53.     01431         MOVE    MON1,MON
54.     01436         MULT   "3",MON

```



```

55. 01451      SUB      "2",MON
56. 01464      RESET    TABLE,MON
57. 01471      GOTO     ERROR IF EOS
58. 01475      MOVE     TABLE,AMON
59. 01502      MOVE     AMON,ADATE
60. 01507      RESET    INDATE,4
61. 01513      LENSET   INDATE
62. 01516      RESET    INDATE,3
63. 01522      RESET    ADATE,4
64. 01526      APPEND   INDATE,ADATE
65. 01533      RESET    INDATE,6
66. 01537      RESET    INDATE,5
67. 01543      RESET    ADATE,10
68. 01547      APPEND   INDATE,ADATE
69. 01554      DISPLAY  "YOUR CONVERTED DATE IN ALPHA IS ",ADATE,*W,*W,*W
70. 01623      GOTO     ASK
71.
72.           . ***** ERROR ROUTINE *****
73.           .
74. 01626      ERROR    DISPLAY "*****NOT CORRECT MONTH*****",*W,*W,*W,*W,*W
75. 01672      BEEP
76. 01673      GOTO     ASK
77.
78.           . ***** END OF JOB *****
79.           .
80. 01676      OUT     DISPLAY "*****END OF DATE CONVERSION PROGRAM*****",*W,*W,*W,*W
81. 01756      BEEP
82. 01757      STOP
83. 01760      STOP

```

String Instructions

Write a program to key in a date and display it back. The date may be keyed in a numeric format (e.g. 101575) or alpha (e.g. OCT 15, 1975). If the date is given in numeric, display it back in alpha. If it is given in alpha, display it back in numeric. In either case, make use of a table of months with the following format:

```
TABLE INIT "JANFEBMARAPRMAYJUNJULAUGSEPOCTNOVDEC"
```

Lab Problem
Datashare

Index Sequential File Processing

1. Using DOS, copy NAMEFILE to a work file.
2. Index the work file on the name field (positions 5 - 24).
3. Write a program in Datashare to:
 - a. Access and display records on the work file by keying in the name.
 - b. Update the work file by making changes to the address and balance.
 - c. Delete records.
 - d. Add new records to the file.

```

1.          . ***** INDEXED SEQUENTIAL LAB - DAY 4 *****
2.          .
3.    00000  ACCT    DIM    4
4.    00007  NAME    DIM   20
5.    00036  STREET  DIM   35
6.    00104  CITY    DIM   20
7.    00133  STATE  DIM   15
8.    00155  ZIP     DIM    5
9.    00165  BAL     FORM  4.2
10.   00176  KEY     DIM   20
11.   00225  FL      IFILE
12.   00257  NSTREET DIM   35
13.   00325  NCITY   DIM   20
14.   00354  NSTATE  DIM   15
15.   00376  NZIP    DIM    5
16.   00406  NBAL    FORM  4.2
17.   00417  ANS     DIM    1
18.          .
19.          . ***** BEGINNING OF PROGRAM *****
20.          .
21.   01401  START   OPEN    FL,"ZPOINTER"
22.          .
23.   01424  ASK     KEYIN    *ES,*P1:11,"A=ADD U=UPDATE D=DELETE L=LOOK E=END: ",ANS
24.   01502          CMATCH   "A",ANS
25.   01506          GOTO     ADDIT IF EQUAL
26.   01512          CMATCH   "U",ANS
27.   01516          GOTO     GETKEY IF EQUAL
28.   01522          CMATCH   "D",ANS
29.   01526          GOTO     GETKEY IF EQUAL
30.   01532          CMATCH   "L",ANS
31.   01536          GOTO     GETKEY IF EQUAL
32.   01542          CMATCH   "E",ANS
33.   01546          GOTO     ENDJOB IF EQUAL
34.   01552          BEEP
35.   01553          GOTO     ASK
36.          .
37.   01556  GETKEY  KEYIN    *ES,*P1:11,"KEYIN NAME: ",KEY
38.          .
39.   01602  READ    READ     FL,KEY;ACCT,NAME,STREET,CITY,STATE,ZIP,BAL
40.   01626          GOTO     BAD IF OVER
41.   01632          DISPLAY  *P1:1,"NAME:      ",NAME,*N,*EL,"STREET:  ",STREET,*N,*EL:
42.   01670          "CITY:      ",CITY,*N,*EL,"STATE:   ",STATE,*N,*EL,"ZIP:
43.   01732          ZIP,*N,*EL,"BALANCE: ",BAL,*N,*EL,"ACCOUNT: ",ACCT
44.   01767          CMATCH   "D",ANS
45.   02001          GOTO     DELETE IF EQUAL
46.   02005          CMATCH   "U",ANS
47.   02011          GOTO     UPDATE IF EQUAL
48.          .
49.   02015  ASK2   KEYIN    *P1:12,*EL,"N=NEXT FUNCTION: ",ANS
50.   02046          CMATCH   "N",ANS
51.   02052          GOTO     ASK IF EQUAL
52.   02056          BEEP
53.   02057          GOTO     ASK2
54.

```

```

55.      . *** DELETE ROUTINE ***
56.      .
57.      02062  DELETE  DELETE  FL,KEY
58.      02067          DISPLAY *P1:12,*EL,KEY," DELETED",*W,*W,*W
59.      02112          GOTO    ASK
60.      .
61.      . *** ADDITIONS TO FILE ***
62.      .
63.      02115  ADDIT   KEYIN   *ES,"NAME:  ",KEY,*N,"STREET:  ",STREET,*N:
64.      02147          "CITY:   ",CITY,*N,"STATE:  ",STATE,*N:
65.      02177          "ZIP:    ",ZIP,*N,"BALANCE: ",BAL,*N:
66.      02227          "ACCOUNT: ",ACCT
67.      02243          TRAP    DUPKEY IF IO
68.      02247          WRITE   FL,KEY;ACCT,KEY,STREET,CITY,STATE,ZIP,BAL
69.      02273          TRAPCLR IO
70.      02276          DISPLAY *P1:12,*EL,KEY," ADDED TO FILE",*W,*W,*W
71.      02327          GOTO    ASK
72.      .
73.      . *** UPDATE ADDRESS AND BALANCE ***
74.      .
75.      02332  UPDATE  KEYIN   *P40:2,"STREET: ",NSTREET,*P40:3,"CITY:   ",NCITY:
76.      02365          *P40:4,"STATE:  ",NSTATE,*P40:5,"ZIP:    ",NZIP:
77.      02425          *P40:6,"BALANCE: ",NBAL
78.      02444          RESET   NSTREET
79.      02450          GOTO    NEWCITY IF EOS
80.      02454          MOVE    NSTREET,STREET
81.      .
82.      02461  NEWCITY RESET   NCITY
83.      02465          GOTO    NEWSTAT IF EOS
84.      02471          MOVE    NCITY,CITY
85.      .
86.      02476  NEWSTAT RESET   NSTATE
87.      02502          GOTO    NEWZIP IF EOS
88.      02506          MOVE    NSTATE,STATE
89.      .
90.      02513  NEWZIP  RESET   NZIP
91.      02517          GOTO    NEWBAL IF EOS
92.      02523          MOVE    NZIP,ZIP
93.      .
94.      02530  NEWBAL  COMPARE "0",NBAL
95.      02543          GOTO    NEWREC IF EQUAL
96.      02547          MOVE    NBAL,BAL
97.      .
98.      02554  NEWREC  UPDATE   FL;*25,STREET,CITY,STATE,ZIP,BAL
99.      02605          MOVE    "L",ANS
100.     02621          GOTO    READ
101.     .
102.     . *** END OF PROCESSING ***
103.     .
104.     02624  ENDJOB  CLOSE    FL
105.     02627          DISPLAY *ES,*P30:5,"***** END OF JOB *****",*W,*W,*W,*W
106.     02671          STOP
107.     .
108.     . ***** INVALID KEY *****

```

PAGE 3

TODD5/TXT:DR0

```
109.
110. 02672 . BAD BEEP
111. 02673 . DISPLAY *P1:12,*EL,KEY," NOT ON FILE - TRY AGAIN",*W,*W,*W
112. 02736 . GOTO ASK
113.
114. . ***** DUPLICATE KEY *****
115. .
116. 02741 DUPKEY BEEP
117. 02742 . DISPLAY *P1:12,*EL,NAME," ALREADY ON FILE; NEW INFO REJECTED",*W,*W
118. 03025 . GOTO ASK
119. 03030 . STOP
120. 03031 . STOP
```

PROGRAM RELEASE/REVISION FORM

PROGRAM NUMBER: C00134

SYMBOL/VER.REV: CTOS 3.2

SHORT DESCRIPTION: Selectric Typewriter Driver

TYPE: 2 cat-rev #2 TYPE 1.1

DATE: November 16, 1972

LISTING: available

SOURCE: available

SUPPORT: 8k version 1 with CTOS

NEW/CHANGE: change

PURPOSE/CHANGES:

This tape is being re-released with no changes to the program TYPE. The only change is to the catalog. The source of TYPE is no longer included on C00134. The source/object tape is P00344.

RELEASOR'S SIGNATURE: 

cc:

Document Control
Technical Publications
Systems Engineering
Quality Control
Shipping
International Marketing