

BASIC/3000

LOG-ON

CARRIAGE-RETURN (ACCESS MPE.)

:HELLO MIKE, ABLE/SNEK.STUDENT

:BASIC

>EXIT

:BYE

CR, CARRIAGE RETURN;

LF, LINE FEED;

CTRL, CONTROL KEY FOR SPECIAL CHARACTERS.

H^C, CONTROL H;

X^C, CONTROL X;

Y^C, CONTROL Y.

POCKET BASIC

SPECIAL CHARACTERS (35).



HP Computer Museum
www.hpmuseum.net

For research and education purposes only.

BASIC/3000

:BASIC

BASIC 257.96

> 10 PRINT 4/3*3.141926*15.3**2

NAME AVER1

10 REM

20 REM READ AND PRINT AVERAGE VALUES.

50 PRINT (A+B+C+D+E)/5

40 READ A,B,C,D,E

05 DATA 1,7,6,3.255,107, "TEST"

ONE CURRENT PROGRAM IN THE USER'S WORK AREA.

STATEMENT NUMBERS, 1-9999, ARE ORDERED BY BASIC.

STATEMENT TYPES REM, PRINT, READ, DATA

EDIT COMMANDS

RENUMBER, REN;

DELETE, DEL;

LENGTH, LEN;

SCRATCH, SCR;

LIST.

POCKET BASIC
COMMANDS (7).

BASIC REFERENCE
ESSENTIAL OF BASIC (2-45).

LIBRARY COMMANDS

NAME

SAVE

GET

APPEND

PURGE

CATALOG, CAT

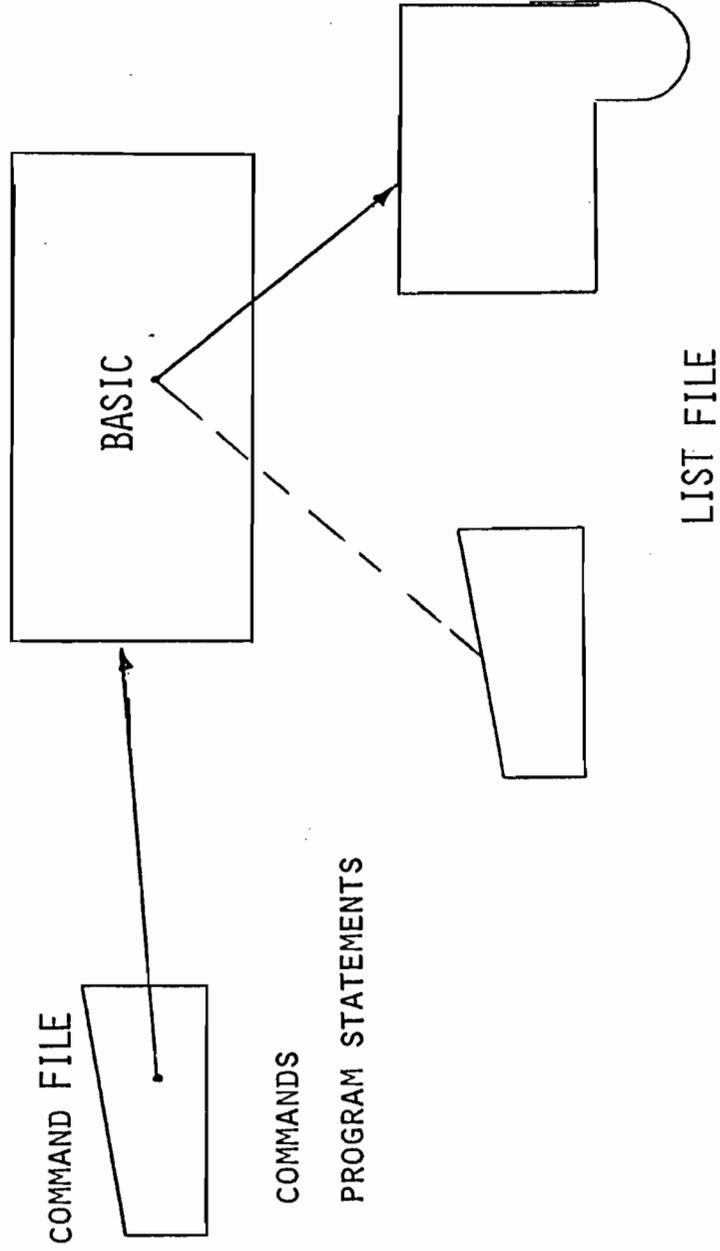
COMMANDS

RUN, EXECUTE COMMAND.

>SYSTEM, SYSTEM COMMAND

:RESUME

BASIC FILES



ASCIIFILE REFERENCE

>SYSTEM

: FILE TUNA; DEV=LP

: RESUME

LIST 1-500,OUT = TUNA

TUNA IS AN ASCIIFILE REFERENCE TO THE LINE PRINTER DEVICE.

POCKET BASIC
COMMAND,LIST(7)

MPE OPERATING SYSTEM REFERENCE SECTION V
MANAGING FILES, :FILE COMMAND (5-10).

WORK SESSION

```
NAME      MOON1
10 REM   VOLUME OF A SPHERE,  $V=4/3 R^3$ 
20 REM   THE AVERAGE RADIUS OF THE EARTH IS 6,371 KILOMETERS
30 REM   THE AVERAGE RADIUS OF THE MOON IS 1,738 KILOMETERS
36 REM   THE VOLUME OF THE EARTH IS  $1,082 \times 10^{21}$  CUBIC METERS
05 REM
04 REM
40 REM    $V(E)/V(M)$ 
50 A = 6371**3/1738**3
60 V = 1/A*1.082E21
55 PRINT "V(E)/V(M)=";A
45 REM V IS THE VOLUME OF THE MOON
70 PRINT "V(M)=";V
```

WORKSESSION CONTINUED

1. GET THE MOON PROGRAM FILE.
2. RENUMBER THE CURRENT PROGRAM.
NEW FIRST IS 20, INCREMENT BY 5.
3. RENUMBER STATEMENTS 55 TO THE END.
NEW FIRST IS 100, INCREMENT BY 20.
4. CREATE AN ASCIIFILE REFERENCE FOR THE LINE PRINTER.
5. LIST THE MOON PROGRAM ON THE LINE PRINTER.
6. PURGE THE OLD MOON PROGRAM FILE AND SAVE THE NEW VERSION.

ESSENTIAL OF BASIC

BASIC REFERENCE

ESSENTIALS OF BASIC (SECTION 2)

NUMERIC EXPRESSIONS

10 REM
20 REM THE AVERAGE RADIUS OF THE MOON IS 1,738 KILOMETERS.
30 R = 1738
40 PRINT 4/3*PIX(R**3)

CONSTANTS

1. 2,3,5,7,9,11,13,17,19
2. 1343.57, 59.67, 183.49
3. 1.035 E+02, 2.57E2, 6.77E-33
4. 3.14159265359L1, 1.41421356237L+1, 1.745 3292520L-2
5. (1.414214,1), (6,7), (3.17, 0.2159)
6. "ABCD", "USER TECHNIQUES", "3 LIVE MONKEYS", '13'10"LIZ"'34

POCKET BASIC

ASCII (BACK COVER)

BASIC VARIABLES

NUMERIC VARIABLES

A THRU Z OR AØ THRU Z9

K, J9, D3

STRING VARIABLES

A\$ THRU Z\$ OR AØ\$ THRU Z9\$

K\$, J9\$, D3\$



SUBSCRIPTED VARIABLES

10 DIM K2(10), J9(10,10) IMPLIED

20 DIM B2(2), D3(13,13), K9\$(6), B\$(5,72)

VARIABLES TYPES

10 DIM B1(10), B2(20,15), B3(9)

20 REAL W4(5), W2(18,3)

70 INTEGER D1, D2, F1(4)

80 LONG P3, Q(6)

100 COMPLEX H1, M2(16,16),N

BASIC BUILT-IN-FUNCTIONS

CPU(X), CPU SECONDS;

SQR(X), \sqrt{X} $\left. \begin{array}{l} \\ \\ \\ \end{array} \right\} \{X \geq 0\}$;

PIX(X), $\pi * X$;

RND(X), PSEUDO-RANDOM - NUMBER

$0 \leq \text{RND}(X) < 1$

POCKET BASIC
FUNCTIONS (31-33)

BASIC REFERENCE
BUILT-IN-FUNCTIONS (APPENDIX E)

MODULE: _____

TOPIC REF: _____

APPLICABLE REF: _____

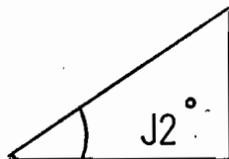
handout

PURPOSE _____

CONVERSIONS FOR THE BUILT-IN FUNCTIONS

ANGLES MUST BE CONVERTED TO RADIANS.

EXP(x) AND LOG(x) USE NATURAL LOGS.



IF J2 IS AN ANGLE IN DEGREES
 THEN $\text{PIX}(J2/180)$, IS THE ANGLE IN RADIANS.

$$360 \text{ DEGREES} = 2\pi \text{ RADIANS.}$$

$$0.01745329 \frac{\text{RADIANS}}{\text{DEGREE}}$$

$$57.29578 \frac{\text{DEGREES.}}{\text{RADIAN}}$$

LOGARITHM

LOG, LOGARITHM BASE 10

LN, LOGARITHM BASE e $e = 2.71828\dots$

$$0.4329448 \frac{\text{LOG}}{\text{LN}} ;$$

$$2.3025851 \frac{\text{LN}}{\text{LOG}} ;$$

DETERMINE THE VALUE 10^{X2} ;
 $\text{EXP}(2.3025851 * X2)$.

INST.

NAME

BUILT-IN-FUNCTIONS

10 PRINT SQR(14), SQR(2600), SQR(13.5)

20 PRINT CPU(Ø)

100 REM CONVERT TO RADIANS

120 REM RADIANS = γ * Z DEGREES
180

140 PRINT PIX (Z/180)

500 PRINT SQR(RND(44))

WORK SESSION

LIST THE BASIC BUILT - IN - FUNCTIONS WHICH WILL BE REQUIRED FOR THESE COMPUTATIONS.

1. COSINE 53 DEGREES;
2. A RANDOM VALUE BETWEEN 1 AND 6;
3. THE NUMBER OF CPU SECONDS A BASIC PROGRAM HAS RUN.

OPERATORS

ARITHMETIC OPERATORS

+ ADD

A + B;

- SUBTRACT

A - B;

* MULTIPLY

A * B;

/ DIVIDE

A / B;

** EXPONENTIATE
(^)

A ** B;

MOD MODULO

A MOD B; (REMAINDER.)

UNARY OPERATORS

+ PLUS,

A ++ B;

- NEGATE

A ** -B;

MOD OPERATOR

$$3 \text{ MOD } 2 = 1;$$

$$(645 \text{ WORDS}) \text{ MOD } \frac{(128 \text{ WORDS})}{\text{SECTOR}} = 5 \text{ WORDS};$$

$$(44 \text{ DAYS}) \text{ MOD } \frac{(7 \text{ DAYS})}{\text{WEEK}} = 2 \text{ DAYS.}$$

MINIMUM AND MAXIMUM OPERATORS

16 MIN 4 = 4

408 MIN -200 = -200

16 MAX 4 = 16

-1632 MAX -1488 = -1488

30 PRINT 5 MIN 6*30

40 PRINT 7 MAX(A MIN B)



BASIC OPERATORS

RELATIONAL OPERATORS

< LESS THAN

= EQUAL

<= LESS THAN OR EQUAL

> GREATER THAN

<> NOT EQUAL
(#)

>= GREATER THAN OR EQUAL

TRUE = 1

FALSE = 0

?
A < B

?
A = B

?
A <= B

?
A > B

?
A <> B

?
A >= B

TRUE

FALSE

RELATIONAL OPERATORS

?
A < B

?
A >= (B*C)

10 IF A2 < B6 THEN PRINT A2

20 IF A2 >= (B6*C8) THEN A2 = A2 + 1

30 IF (A2 + C8) = (B6 + J9) THEN A2 = J9

40 IF A2 > B6 OR C8 > B6 THEN PRINT B6

STRING OPERAND

+ CONCATENATE, A\$ + B\$

"GREEN" + "HOUSE" = "GREENHOUSE"

'34 "UP" + "DOWN" + '34 = "UPDOWN"

POCKET BASIC;

ASCII TABLE (BACK COVER).

EXPRESSION TYPES

100 INTEGER G9, C6

200 LONG B4, M2

300 COMPLEX T2, W5

400 PRINT G9 + B4

LONG EXPRESSION

420 W5 = T2*C6

COMPLEX EXPRESSION

440 T2 = 1.4L1 * (13.6,4.2) COMPLEX EXPRESSION

BASIC REFERENCE

NUMERIC EXPRESSIONS (4-6)

ASSIGNMENT STATEMENT

10 C = C+1
20 F2 = A*-B
40 U = ATN(EXP(M*T4))
60 A1 = K5 = L3 = S7 = 0
80 B7 = 4, T3 = 9, J2 = 6
90 P2\$ = "PRICE 1", P3\$= "PRICE 4", S2 = T4 = 7

BASIC REFERENCE
STATEMENTS (2-10)

AUTOMATIC TYPE CONVERSION

10 INTEGER P,Y

20 REAL I,J

30 COMPLEX M1, T2

60 $P = I * J$

70 $Y = M1 + 14.7L0$

BASIC REFERENCE

TYPE CONVERSION TABLE (4-7)

POCKET BASIC
CONVERSION TABLE (16)

WORK SESSION

WRITE BASIC STATEMENTS

1. INITIALIZE THIS LIST OF VARIABLES TO ZERO; A1, C5, B2, K8, M6

2. INITIALIZE THESE VARIABLES $K \leftarrow 5$, $J6 \leftarrow 3$, $L2 \leftarrow 33$.

$$3. R = \frac{A + BX}{C + DX}$$

$$4. B = \frac{-1 + A^2}{2X} \quad 4X$$

$$5. I = \frac{X - 1}{A} \quad \frac{1}{AP} \quad \text{LOG}(A + B e^{PX})$$



READ, DATA, RESTORE, REM

60 REM

70 READ A,B,C,D

80 DATA 5.6L+Ø, (16.55,33),4,6, "BEEF", "LOG"

90 DATA 17, 19, 23, 29

100 RESTORE

110 RESTORE 90

120 READ D5(1), D5(2), D5(3), D5(4)

PRINT STATEMENT

```
150 PRINT "A", "B", "C5", "N6", "T2"  
160 PRINT A,B,C5,N6,T2  
170 PRINT  
180 PRINT F6(1); F6(2); F6(3); F6(4); F6(5)  
190 PRINT D1(11), D1(12),  
200 PRINT D1(13), D1(14)
```


PRINT STATEMENT

COLUMNS

COLUMN	COLUMN	COLUMN	COLUMN	COLUMN
1	2	3	4	5
A	B	C5	N6	T2
-4300	-27	-1.007	-9567	-999
2.345E+6	93	2.006	1067	1000
-20	800	56.863	.0067	88

PRINT STATEMENTS
PACKED FIELDS

1.	2	-3	4	5	13	99		
2.	1000		990087		32678	20034		
3.	1.007		-67.32		.00084	.87654		
4.	-2.34568E+06						15E-32	
5.	ABCDE							
6.	A		B					C

BASIC REFERENCE
NUMERIC OUTPUT FORMATS (2-31)
VARIABLE TYPES (SECTION 4)

POCKET BASIC
PRINT STATEMENT (15-17)

PRINT FUNCTIONS

1. TAB(2), TAB(65), TAB(15), TAB(82)
2. SPA(44), SPA(3)
3. LIN(6), LIN(2), LIN(18)
4. LIN(-4), LIN(-6)
5. CTL(32), SINGLE SPACE;
6. CTL(48), DOUBLE SPACE;
7. CTL(49), PAGE EJECT;
8. CTL(45), TRIPLE SPACE

MPE REFERENCE

FIGURE 6-1. CARRIAGE-CONTROL DIRECTIVES

PRINT FUNCTIONS

```
10 PRINT  
20 PRINT LIN(1);  
30 PRINT LIN(1)  
40 PRINT "HERE"; TAB(15)  
50 PRINT SPA (10), A, SPA(10), LIN (-5)  
60 PRINT CTL(49)  
70 LIN (7)
```



ERROR#2

INVALID STATEMENT

laboratory project

MODULE:

TOPIC REF:

APPLICABLE REF:

OBJECTIVE	_____

LAB ASSIGNMENT

PART 1

FIELD1

FIELD1 Demonstrates the default print formats for small integers, large integers and fixed point decimal fractions.

PROBLEM

Run FIELD1 and determine what the default field sizes are.

- 1.1 Small integer size?
- 1.2 Large integer size?
- 1.3 Fixed point size?

Program FIELD1

1. DATA array G5.
STRINGS T1\$, T2\$.

2. Initialize the print scale
0123456789... 0128456789

3. PRINT SCALE AND DATA

Execute step 3 until the data runs out.

4. Data

Small integers values
Large integer values
fixed point decimal fractions

NAME _____

INST. _____

```

LIST
FIELD1
10 DIM G5E3]
20 DIM T1S172],T2S172]
30 REM
40 REM PRINT SCALE
50 T1S="0123456789"
60 T2S=T1S+T1S
70 T2S=T2S+T2S+T2S+T1S
80 REM
90 REM PRINT SCALE AND DATA
100 MAT READ G5
110 PRINT LIN(1)
120 PRINT T2$
130 PRINT G5E1];G5E2];G5E3]
140 GOTO 90
150 REM
160 REM SMALL INTEGER VALUES
170 DATA 20,-300,999
180 REM
190 REM LARGE INTEGER VALUES
200 DATA 990087,-32678,20034
210 REM
220 REM FIXED POINT DECIMAL FRACTIONS
230 DATA 1.007,-67.32,.0084

```

PURPOSE _____

handout

APPLICABLE REF: _____

TOPIC REF: _____

MODULE: _____

laboratory project

MODULE: _____

TOPIC REF: _____

APPLICABLE REF: _____

OBJECTIVE _____

LAB ASSIGNMENT

Part 2

VALUES

1	-15.233
2	10
3	6.5
4	"COCK FIGHT"

1. Print the values in 4 columns

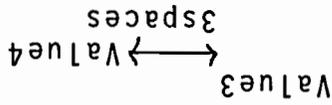
2. Print the values in 4 columns, that are 12 characters wide.

3. Value1
Value2

DOUBLE SPACE

Value 3 → 7 spaces → Value 4

4. Value1
Value2
Value3



5. NUMBERS "STRINGS"

Value1
Value2
Value3

Value4

NOTE: "QUOTE mark is an ascii (34)

EQUIPMENT COST AND DEPRECIATION

ITEM	NUMBER	MATERIAL COST	DEPRECIATION
SUPPLY CARS	16	40,000	10 YEARS
STPR CNTR SPARS	3	18,000	10 YEARS
ROCK DSTR	3	16,200	10 YEARS

PRINT USING
FORMAT STRINGS

16A

8AXX6A

2(D,DDXX,4A)

4D,DDX//DX4D,DDD

KX, "BUSHELs"

12(11A, \$5D,DDXX)

#,3D,DD

PRINT USING
FORMAT STRINGS

CARRIAGE CONTROL

+, SUPPRESS LINE FEED;
-, SUPPRESS CARRIAGE RETURN;
#, SUPPRESS CARRIAGE RETURN
AND LINE FEED

FORMAT ELEMENTS

A, ASCII CHARACTER;
D, DECIMAL DIGIT;
X, BLANK CHARACTER;
, DECIMAL POINT;
K, KRUNCH FORMAT;
/, SEPARATOR GENERATES
A CARRIAGE RETURN LINE FEED;
, FIELD SEPARATOR.

POCKET BASIC
PRINT Using (18)
BASIC REFERENCE
FORMATTED OUTPUT (SECTION 9)

FORMAT SYMBOLS

A,	ASCII CHARACTER;	AXA,15A
D,	DECIMAL DIGIT;	DDD
M,	MINUS SIGN (FIXED POSITION);	MDD, DDM
S,	SIGN CHARACTER (MAY FLOAT);	SDDD,D, DSXDD
X,	BLANK CHARACTER;	5X
.	DECIMAL POINT;	D.D, X.DD
/,	FIELD SEPARATOR WHICH GENERATES A CARRIAGE RETURN LINE FEED;	AA//D
,	FIELD SEPARATOR;	,,A,, DD
C,	COMPLEX FORMAT;	C(3D,M4D,D)
E	SCIENTIFIC NOTATION	DD,DEDD
I,	GENERATE THE EDITING SYMBOL I IMAGINARY;	DDXI
\$,	GENERATE A \$ EDITING SYMBOL;	\$DD
K,	KRUNCH FORMAT	"JAN.",XK," , 1984"
+ OR -,	SEPERATE THE REAL AND IMAGINARY PARTS OF A COMPLEX FORMAT	DD+DDI, D + D.DDDI
CARRIAGE CONTROL		
+	SUPPRESS LINE FEED;	+ ,6(4D.)
-	SUPPRESS CARRIAGE RETURN;	- ,16A
#	SUPPRESS CARRIAGE RETURN AND LINE FEED	# , 6A,XX,/,/, 2A

PRINT USING

106 PRINT USING "DDD.DD"; 105.2, 67.8, -12, 1080.56

120 PRINT USING B8\$; D,E,F

130 PRINT USING 140; "EXPENSE△TABLE", LIN(1), E1, E2, E3

140 IMAGE 14A, 2X, 54D.DD, "COST"

200 PRINT USING "BEANS-▲", KX, "BAGS"; P, LIN(1), P2

205 PRINT USING C8\$; TAB(40), A,T,G,R

100 C8\$ - "+, 5D.D, 3(2D,XX)"



GOTO, END, STOP

GOTO STATEMENT NUMBER.

END SIGNAL PROGRAM COMPLETION.

STOP TERMINATE.

80 GOTO 70

90 GOTO 800

100 END

BASIC REFERENCE
ESSENTIALS (SECTION 2)

POCKET BASIC
STATEMENTS (9)

COMPUTED GOTO

20 GOTO N OF 40, 90, 100, 120

30 END

N GOTO

- 1 40
- 2 90
- 3 100
- 4 120

OTHERWISE EXECUTE 30

Page 30

IF THEN ELSE

```
10 IF T2 < (N-1) THEN D = Y**2
20 IF M8 > H2 THEN K = K + 1 ELSE K = 0
30 IF J > T(5) THEN GOTO 90
40 IF S = 44 THEN 400
```

EXPRESSION <> 0, TRUE

EXPRESSION = 0, FALSE

IF THEN ELSE

230 IF M>0 THEN P = PI X(1/2) *EXP(-M)

240 ELSE DO

250 IF M = 0 THEN P = 0

260 ELSE P = PIX(1/2)*EXP(M)

270 DOEND

WORK SESSION

WRITE BASIC PROGRAMS

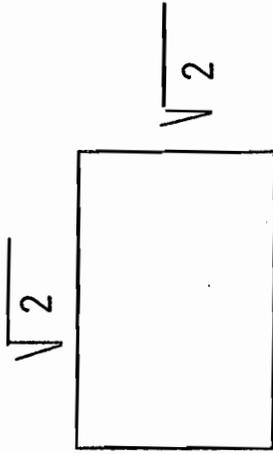
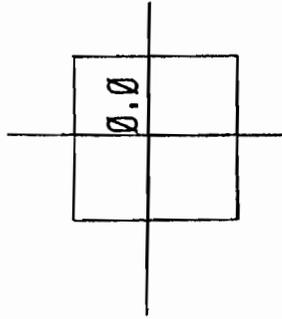
1. IF $A > B$, SET X TO 16.9;
OTHERWISE Y = 23.1.

2. READ N

PRINT TEST CONDITION ONE FOR $N = (1,2,8)$;
PRINT TEST CONDITION TWO FOR $N = (4,5,6)$;
PRINT END-OF-PROGRAM FOR $N = (3,7)$;
OTHERWISE PRINT INVALID VALUE, $N=?$

WORK SESSION

3.



<u>A1</u>	<u>B1</u>	P(A1,B1)
-0.3	0.62	YES

READ A1 AND B1 AND DETERMINE IF THE POINT IS IN THE SQUARE. PRINT A MESSAGE AS ABOVE.

GOSUB RETURN

30 GOSUB 100

100 C = C + A + B

110 PRINT A,B,C

120 A = B = 0

130 I = I + 1

140 RETURN

40 A = A + 1

50 B = B + 1

GOSUB RETURN

50 GOSUB N OF 100, 200, 300

60 END

<u>N</u>	<u>SUBROUTINE</u>
----------	-------------------

1	100
---	-----

2	200
---	-----

3	300
---	-----

OTHERWISE EXECUTE 60



FOR NEXT

10 FOR I = 1 TO 200

20 PRINT "I = "; I, A(I), B(I),

30 PRINT A(I) + B(I)

40 NEXT I

90 A = 0

100 FOR W = 0 TO 1 STEP 0.1

120 A = A + I

130 PRINT "I="; I, "A="; A

140 NEXT W

READ PRINT

10 PRINT (FOR J=1 TO 10, C[J], T[J])

30 READ (FOR R5=1 TO N, (FOR Q2=1 TO M, C[R5, Q2]))

T4

```
10 REM
20 REM SADDLE POINT FOR A 4*4 ARRAY.
30 REM S(4,4), ARRAY.
40 REM M1(J), MAXIMUM VALUE IN COLUMN J.
50 REM M2, MINIMUM (COLUMN MAXIMUM).
60 REM M3, ROW MINIMUM.
70 REM IF T=0, NO SADDLE POINT (TRUE).
80 REM IF T=1, FOUND A SADDLE POINT (FALSE).
90 REM I, ROW INDEX.
100 REM J, COLUMN INDEX.
110 REM IMPLICIT DIM S(10,10),M1(10).
120 REM
130 REM
140 MAT READ S(4,4)
150 MAT PRINT LIN(1),S;
160 REM INITIALIZE THE COLUMN MAXIMUMS.
170 FOR J=1 TO 4
180   M1[J]=S[I,J]
190 NEXT J
200 REM
210 REM FIND THE COLUMN MAXIMUMS.
220 FOR I=2 TO 4
230   FOR J=1 TO 4
240     M1[J]=M1[J]-MAX S[I,J]
250   NEXT J
260 NEXT I
270 REM
280 REM INITIALIZE M2.
290 M2=M1[1]
300 REM
310 REM FIND THE SMALLEST COLUMN MAXIMUM.
320 FOR J=2 TO 4
330   M2=M1[J]-MIN M2
340 NEXT J
350 REM
360 REM SET T:= FALSE.
370 T=0
380 REM
390 REM IS THIS ROW MINIMUM A SADDLE POINT?
400 REM
410 REM IF THE ROW MINIMUM IS EQUAL TO
420 REM SMALLEST COLUMN MAXIMUM. THAT IS A SADDLE POINT.
430 REM
440 REM INITIALIZE M3, THE ROW MINIMUM
450 REM M3=S(1,1)
460 REM
470 REM
480 REM SET M3.
490 REM M3 IS THE MINIMUM VALUE IN ROW I.
500 FOR I=1 TO 4
510   M3=S[I,1]
520   FOR J=2 TO 4
530     M3=M3 MIN S[I,J]
540   NEXT J
550   REM
560   REM TEST
570   REM IS M3 A SADDLE POINT.
580   IF M3=M2 THEN DO
590     T=1
```



```
600 GOTO 640
610 DOEND
620 NEXT I
630 REM
640 REM PRINT THE RESULTS
650 IF NOT T THEN DO
660 PRINT "THIS ARRAY HAS NO SADDLE POINT",LIN(2)
670 DOEND
680 ELSE DO
690 PRINT USING 700;M3,1
700 IMAGE /"THE SADDLE POINT IN THIS ARRAY HAS A VALUE OF",XKX,"IN R&
OW",XK,"."//
710 DOEND
720 REM
730 REM CONTINUE?
740 READ C
750 IF C THEN 140
760 PRINT LIN(2),"FINI"
770 DATA 1,17,16,33
780 DATA 5,6,7,8
790 DATA 9,10,-1,11
800 DATA 18,13,14,15
810 DATA 1
820 DATA 1,2,3,4
830 DATA 5,6,7,8
840 DATA 9,10,-1,11
850 DATA 12,13,14,15
860 DATA 0
```

laboratory project

MODULE: _____

TOPIC REF: _____

APPLICABLE REF: _____

OBJECTIVE _____

LAB PROBLEM

1. ARRAY C1 and C5
DIM C1(4,4) and C5(4,4)
SET T=1 if all the elements of C1 are greater than the
elements of C5. Otherwise set T=0.

2. The sum of the years' digits methods of depreciation

N is the assets life in years.
S is the sum of the digits 1,2,...,N.
The sum S is the denominator of a fraction which
determines the depreciation rate. The numerator
of the fraction is N the first year, N-1 the second
year, N-2 the third year...

3 year asset life;
S = 1 + 2 + 3
S = 6
Year 1 = 3/6 3/6
Year 2 = (3-1)/6 2/6
Year 3 = (3-2)/6 1/6

Print a depreciation table for an asset life of 8,16,20 years.

INST.

NAME

MODULE: _____

TOPIC REF: _____

APPLICABLE REF: _____

laboratory project

OBJECTIVE _____

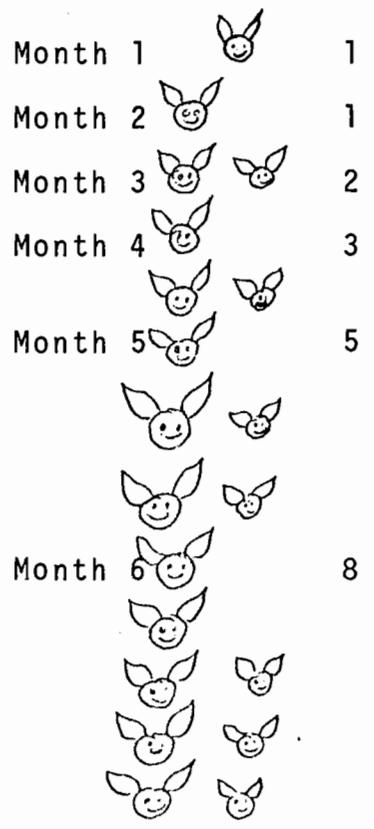
LAB PROBLEM

3. How many pairs of rabbits can be produced from a single pair of rabbits in one year's time.
 - 3.1 Each pair of rabbits produces a new pair of offspring each month.
 - 3.2 Each new pair become fertile after one month.
 - 3.3 Rabbits never die.

This is the definition of the Fibonacci series.

$$0, 1, F_{n-2} + F_{n-1}, \dots$$

0, 1, 1, 2, 3, 5, 8, 13, 21, 34



INST.

NAME

BASIC DEBUGGING

TRACE A2, W4, M6, D1\$

TRACE M6(5), S2(4,3)

TRACE P(*), T(*,*), U\$(*,*)

TRACE PIX,FNA,FNB

TRACE 10, 80, 200-300

TRACE PROG

TRACE

POCKET BASIC
DEBUGGING (38)

BASIC REFERENCE
DEBUGGING (SECTION 7)

BREAK

>BREAK 100,200, 300-500

>BREAK PROG

>BREAK

>YC

>UNBREAK 200

> UNBREAK

DEBUGGING

TERMINATE

> ABORT

> GO

> RESUME

> GO 400

DEBUGGING

>SHOW L2, G4, Z1

>SHOW I(4), D6(5,5)

>SHOW R2(*), Y4(*,*), C5\$(*)

>SET E7 = 9.32

>SET M8\$= "GOLDEN"



	RUN ONLY	BROKEN	NORMAL (NOT BROKEN)		RUN ONLY	BROKEN	NORMAL (NOT BROKEN)
ABORT	X	X		RENUMBER			X
APPEND			X	RESUME (GO)	X	X	
BREAK		X	X	RUN	X		X
CALLS		X		SAVE			X
CATALOG	X	X	X	SCRATCH	X		X
CREATE	X	X	X	SET		X	
DELETE			X	SHOW		X	
DUMP	X	X	X	SPOOL	X	X	X
EXIT	X	X	X	BASIC STATEMENTS			X
FILES		X		SYSTEM	X	X	X
GET	X		X	TAPE	X	X	X
KEY	X	X	X	TRACE		X	X
LENGTH		X	X	UNBREAK		X	X
LIST		X	X	UNTRACE		X	X
NAME			X	XEQ	X	X	X
PUNCH		X	X				
PURGE	X	X	X				

DEBUGGING

100 FOR I=3 TO 72 STEP 5

120 FOR J=N-1 TO 1 STEP-1

130 GOSUB 500

140 GOSUB 600

150 NEXT J

160 NEXT I

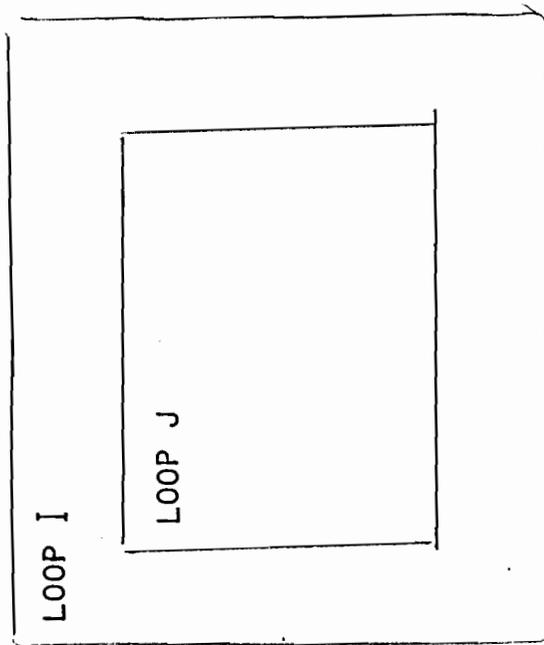
>TRACE J,A,B,C

>TRACE 500, 600

>BREAK 150

WORK SESSION

DON KNUTH'S STRAIGHT INSERTION SORT



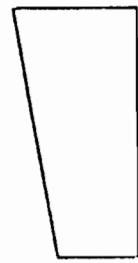
1. PRINT THE INITIAL CONDITION OF ARRAY S.
2. TRACE THE VALUES OF J.
SUSPEND THE PROGRAM WHEN
YOU COMPLETE THE J LOOP.
3. TRACE THE INDEX I, TEMPORARY X AND THE ARRAY S.
4. CONTINUE PROCESSING.

```
05 REM
10 REM D. KNUTH VOLUME 3
20 REM STRAIGHT INSERTION SORT.
30 REM S(16), DATA ELEMENTS.
40 REM X, TEMPORARY.
50 REM I, MAJOR INDEX
60 REM J, MINOR INDEX
65 DIM S(16)
70 READ (FOR I = 16, S(I))
80 N = 16
90 REM
100 FOR I = 2 TO N
110 X = S(I)
120 FOR J = I - 1 TO 1 STEP -1
130 IF X > S(J) THEN 160
140 S(J + 1) = S(J)
150 NEXT J
160 S(I) = X
170 NEXT I
180 PRINT (FOR I = 1 TO 16, S(I), LIN(I))
190 REM
200 DATA 503, 87, 512, 61, 908, 170, 897, 275
210 DATA 653, 426, 154, 509, 612, 677, 765, 703
```

INPUT STATEMENT

10 INPUT A,B,C\$

INPUT FILE



\$STDIN

? 15, 512L1, "MOUSE", CAT

? FIG. A 15 INCHES BY 45 INCHES

POCKET BASIC
INPUT STATEMENT (11)

BASIC REFERENCE
INPUT STATEMENT (2-38)

INPUT STATEMENT

20 INPUT "K2? ", K2

30 INPUT "A ", A, "T? ", T

40 INPUT S, "T6? ", T6, "U? ", U

60 INPUT "NEW VALUE Z?=", Z, (FOR I = 1 TO 10, Z[I])

INPUT

>BASIC

```
10 REM D6(10), ANGLES IN RADIAN.  
20 INPUT (FOR I = 1 TO 10, "D6? Δ", D6(I))
```

D6?>BASIC

```
>> 10 REM D(10) ANGLES IN DEGREES.  
20 INPUT (FOR I=1 TO 10, D(I))  
30 FOR I = 1 TO 10  
40 D(I) = PIX(D(I))/180  
50 NEXT I  
60 PRINT (FOR I = 1 TO 10, D(I), LIN(1))  
>>EXIT
```

??

BUFFERED INPUT

10 INPUT A,B,C ?10,15,20,25,30

20 INPUT A,B,C: 25,30

1. INPUT :

2. INPUT A:

3. INPUT :A

4. INPUT :A:

BUFFERED INPUT

10 INPUT B1,B2,B3:

20 INPUT :B4,B5

100 INPUT :

130 INPUT :A:

BUFFERED INPUT

300 INPUT :

310 INPUT :P:

320 PRINT P

330 IF P = Ø THEN END

340 GOTO 310

BUF(X)

785, 2.2, "CRUEL", PLEASANT

INPUT :

BUF(x)	ITEM
1	REAL
2	STRING
4	NO DATA IN THE BUFFER
5	INTEGER
6	LONG
7	COMPLEX

WORK SESSION

DIM D1(10), D2(10)

INPUT DATA ITEMS

1. PRINT THE NEXT ITEM IF IT IS A STRING.
2. STORE THE NEXT ITEM IN D1 IF IT IS AN INTEGER VALUE.
3. STORE THE NEXT ITEM IN D2 IF IT IS A REAL VALUE.
4. STORE THE NEXT ITEM IN A DUMMY VARIABLE IF IT IS LONG OR COMPLEX.
5. AFTER INPUTTING 10 INTEGERS AND 10 REAL VALUES PRINT D1 AND TERMINATE.

STUDENT HANDOUT

BUFINI

```
10 REM A0, DATA.
20 REM A1, SUBTOTAL.
30 REM A2, TOTAL.
40 A1=A2=0
50 INPUT :
60 IF BUF(0)=4 THEN DO
70 REM FINI
80 REM TOTAL LINE
90 PRINT "TOTAL= ";A2+A1
100 END
110 DOEND
120 ELSE DO
130 IF BUF(0)=2 THEN DO
135 REM A STRING BEGINS EACH NEW SUBTOTAL.
140 INPUT :A2$:
150 PRINT "SUBTOTAL ";A2$,LIN(1),A1
160 A2=A2+A1,A1=0
170 DOEND
180 ELSE DO
190 REM NUMERIC DATA
200 INPUT :A0:
210 A1=A1+A0
220 DOEND
230 DOEND
240 GOTO 60
250 REM TEST THE NEXT DATA ITEM.
```

BASIC CHARACTER STRINGS

1. "MICE", "QUANTITY DISCOUNT"
2. '29'31 "CRT, HOME UP", "ERASE EOF"
3. '34 QUOTE '34
4. DAT\$(1,27)
5. C2\$, M3\$
6. Z3\$(4), T9\$(17)

POCKET BASIC
ASCII TABLE(BACK COVER)



MODULE: _____

TOPIC REF: _____

APPLICABLE REF: _____

handout

PURPOSE _____

HP2800 CRT CURSOR CONTROL CHARACTERS

- '07, BELL;
- '24, CURSOR RIGHT;
- '25, CURSOR LEFT;
- '26, CURSOR UP;
- '11, CURSOR DOWN;
- '28, HOME DOWN;
- '29, HOME UP;
- '30, ERASE END-OF-LINE;
- '31, ERASE END-OF-FILE;
- '127, RUB-OUT.

INST.

NAME

WORKSESSION

DATE STRING

D8\$

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

DAY	,▲	MONTH	▲	DAY	,	▲	YEAR	,	▲	HOUR	:	MIN.	▲	AM	PM
-----	----	-------	---	-----	---	---	------	---	---	------	---	------	---	----	----

DAY OF THE WEEK (SUN, MON, TUE, WED, THU, FRI, SAT)

MONTH OF THE YEAR (JAN, FEB, MAR, APR, MAY, JUN, JUL, AUG, SEP, OCT, NOV, DEC)

DAY OF MONTH (0-31)

YEAR (1982)

HOUR (1-12)

MINUTE (0-59)

AM OR PM

WORK SESSION

SUBSTRINGS

GIVEN THE DATE STRING D8\$ USE THE NECESSARY SUBSTRINGS.

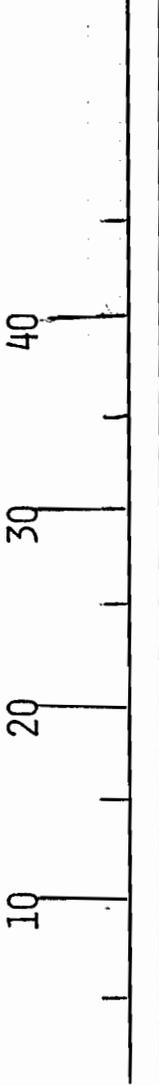
1. PRINT THE ENTIRE DATE STRING. *D8\$*

2. PRINT HOUR:MIN AM OR PM. *D8\$(20)*

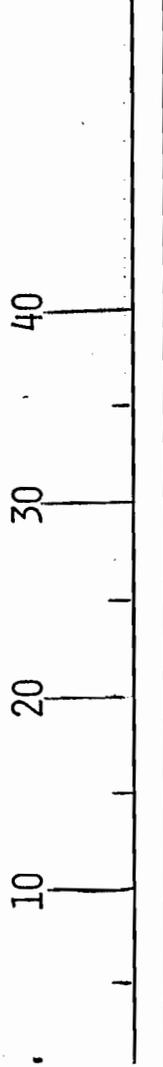
3. PRINT MONTH, DATE AND YEAR. *D8\$(6,8)+D8\$(10,11)+D8\$(14,17)*

4. PRINT DAY OF THE WEEK AM OR PM. *D8\$(10,11)+D8\$(6,27)*

STRING VARIABLES



P6\$ BASIC SIXTH EDITION WITH CORRECTIONS



Z3\$

- 1 BASIC BASIC
- 2 AN INTRODUCTION TO COMPUTER PROGRAMMING
- 3 IN BASIC LANGUAGE
- 4 JAMES S COAN
- 5 GERMANTOWN FRIENDS SCHOOL

SUBSTRINGS



P6\$ BASIC SIXTH EDITION WITH CORRECTIONS

- 1 P6\$ (34) ONS
- 2 P6\$ (25) CORRECTIONS
- 3 P6\$ (21,24) WITH
- 4 P6\$ (7,19) SIXTH EDITION
- 5 P6\$ (21;16) WITH CORRECTIONS
- 6 P6\$ (7;3) SIX
- 7 P6\$ (12;1) BLANK
- 8 P6\$ = P6\$ (1;Ø) NULL
- 9 P6\$ = P6\$ (36,35) NULL

*START R
CHAR. →*

ARRAY SUBSTRINGS

10 20 30 40 50

Z3\$

1 BASIC BASIC
 2 AN INTRODUCTION TO COMPUTER PROGRAMMING
 3 IN BASIC LANGUAGE
 4 JAMES S COAN
 5 GERMANTOWN FRIENDS SCHOOL

1	Z3\$(1)	BASIC BASIC
2	Z3\$(4,9)	COAN
3	Z3\$(5,12,18)	FRIENDS
4	Z3\$(3,4;5)	BASIC
5	Z3\$(2,4;12)	INTRODUCTION
6	Z3\$(2,1;1) + Z3\$(1,7,11) + Z3\$(5,12)	

A BASIC FRIENDS SCHOOL

STRING FUNCTIONS

VALUE RET = STRING
BASE 10 STRING ——— **DECIMAL EQUIVALENT**
OF ASCII CHAR

- 1 CHR\$(A)
- 2 NUM\$(S)
- 3 LEN\$(S)
- 4 UPS\$(S)
- 5 DEB\$(S)
- 6 POS(A\$,B\$)
- 7 WRD(A\$,B\$)
- 8 ROW\$(S)
- 9 COL\$(S)
- 10 DAT\$(N,M)

UPPER CASE
DELETE BLANKS - LEADING & TRAILING

LOOK FOR B\$ WITHIN A\$

POCKET BASIC
FUNCTIONS

BASIC REFERENCE
STRINGS SECTION 5

STRING FUNCTIONS

100 FOR P2 = 0 TO 25

110 R3\$ = CHR\$(P2+65)

120 PRINT R3\$

130 PRINT NUM(R3\$)

140 NEXT P2

200 IF LEN(S3\$) = 0 THEN END

220 L3\$ = UPS\$(Y4\$) → *SHIFT TO ALL UPPER CASE.*

230 J2\$ = DEB\$(" " BLANKS " ")

STRING FUNCTIONS

400 D2\$ = "APPLES ORANGES CHERRIES PLUMS PEACHES"

420 G8 = POS(D2\$, "ERR")

430 T9 = WRD(D2\$, "PLUMS")

500 DIM R2\$(20,72)

510 INPUT "SIZE?", H2

520 REDIM R2\$(H2)

530 PRINT ROW(R2\$), COL(R2\$)

(Find starting point of

size of each string

size of element

*Find
1st
2nd
3rd
4th
5th
6th
7th
8th
9th
10th
11th
12th
13th
14th
15th
16th
17th
18th
19th
20th*

REPEAT

size of element

start

end

Element

Row

Col

DATE\$(N,M)

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

DAY	,	△	MONTH	△	DAY	,	△	YEAR	,	△	HOUR	:	MIN.	△	AM	PM
-----	---	---	-------	---	-----	---	---	------	---	---	------	---	------	---	----	----

DAY OF THE WEEK (SUN, MON, TUE, WED, THU, FRI, SAT)

MONTH OF THE YEAR (JAN, FEB, MAR, APR, MAY, JUN, JUL, AUG, SEP, OCT, NOV, DEC)

DAY OF THE MONTH (0-31)

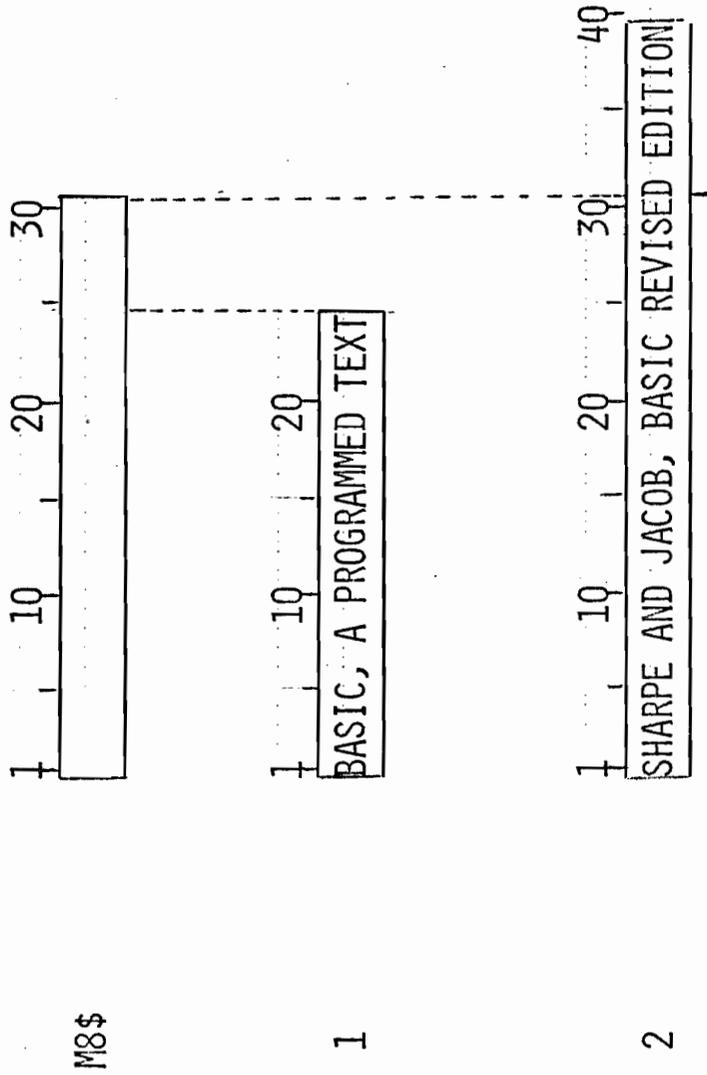
YEAR (1982)

HOUR (1-12)

MINUTE (0-59)

AM OR PM

ASSIGNMENT STATEMENT I



Handwritten note:
w
ALL WORKSHEET #1
T6

ASSIGNMENT STATEMENT II

1 BASIC, A PROGRAMMED TEXT 30

Y1\$

10 BASIC

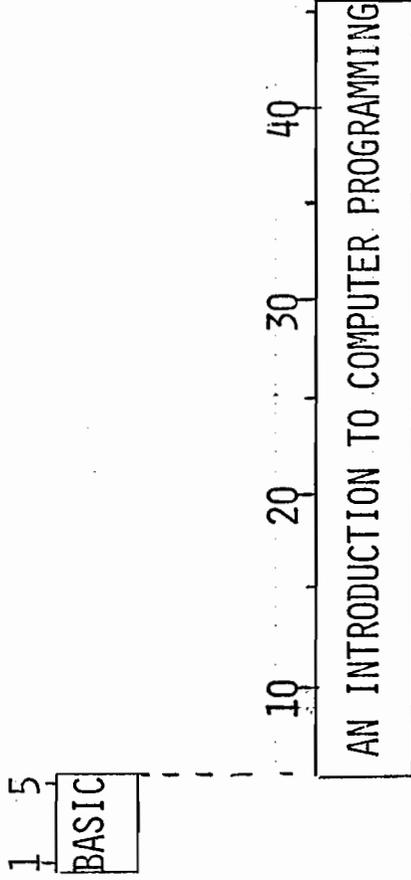
Y1\$(6) = "basic" →

10 20 30 40
PROGRAMMING BY KEMENY AND KURTZ

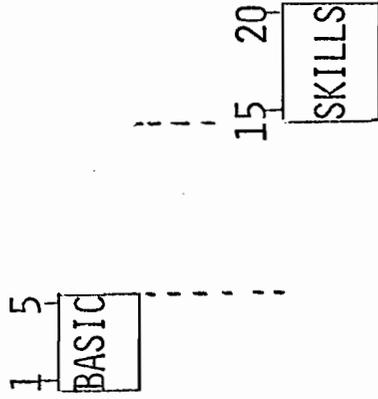
Y1\$(6)

Handwritten note: 1 KEMENY AND KURTZ

ASSIGNMENT STATE II

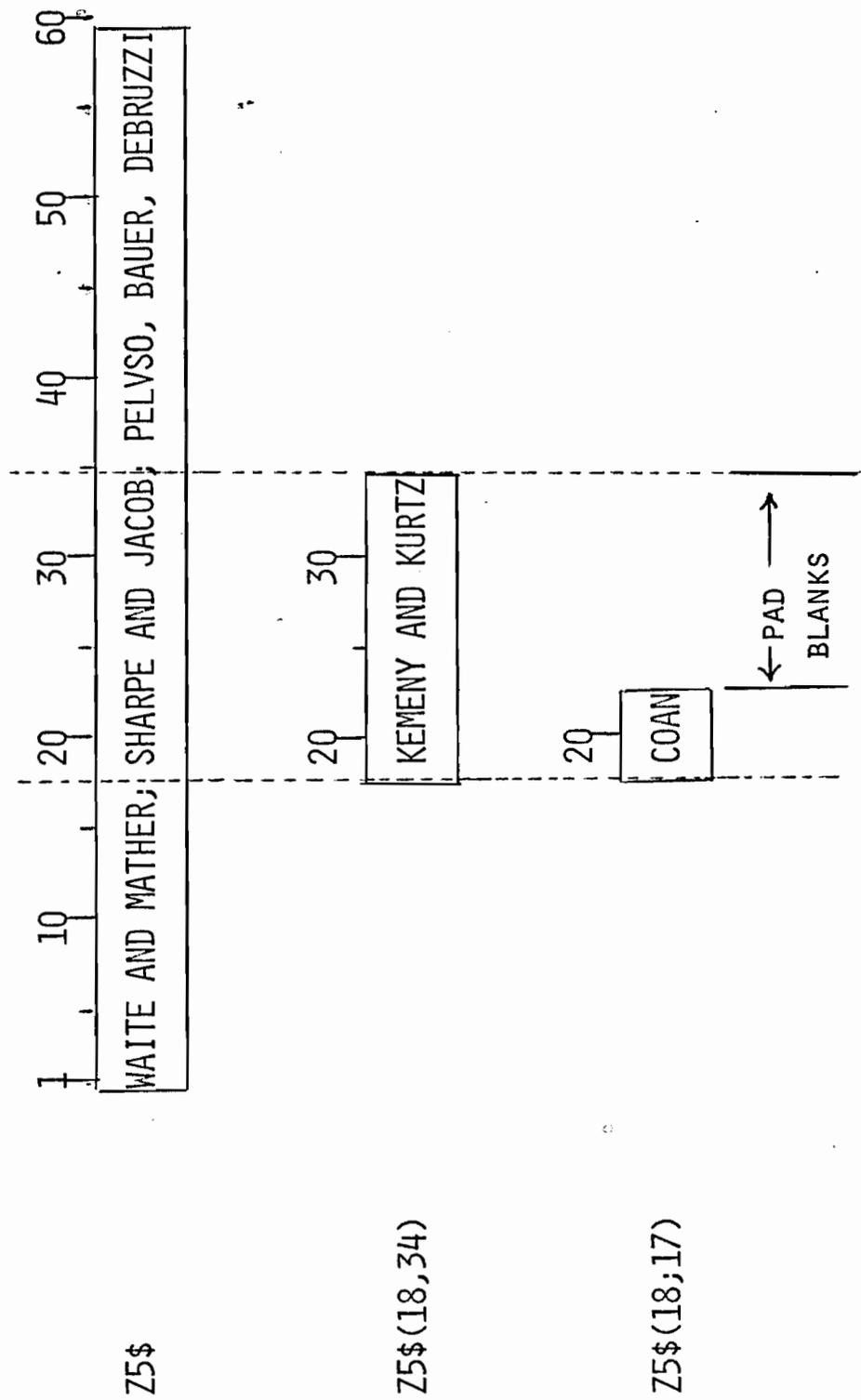


1. W5\$(6)



2 W8\$(15)

ASSIGNMENT STATEMENT III



Z5\$

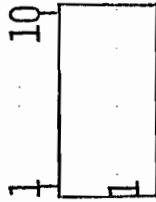
Z5\$(18,34)

Z5\$(18;17)

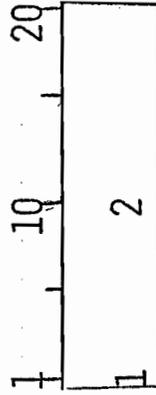
ASSIGNMENT STATEMENT III

PAD WITH BLANKS

Z5\$(1;10) = "1"



Z5\$(10;10) = "2"



STRING COMPARE

< , LESS THAN
= , EQUAL
<= , LESS THAN OR EQUAL

> , GREATER THAN
<> , NOT EQUAL
>= , GREATER THAN OR EQUAL

S1\$> S2\$

ASCII CODES. LEXICOGRAPHIC (ALPHABETIC) ORDER

\$%&
() '95'100
1 2 3 4 5

A
ACD
ACDE
G
HI
ZEBRA

LINPUT

LINPUT S\$ → SAVE AS INPUTTED

EXACTLY

```
10 DIM A$ (72)
20 LINPUT A$
30 IF LEN(A$) THEN END
40 PRINT A$
50 GOTO 20
```

WORK SESSION

WRITE A BASIC PROGRAM

LINPUT TWO STRINGS A\$ AND B\$.

INPUT N AND M.

SET A\$ AND B\$ SO THAT THE SUBSTRINGS

A(N;M) > B$(N;M)$ FOR ANY ARBITRARY VALUES

OF A\$ AND B\$.

WORK SESSION

```
10 DIM A$(72), B$(72), T$(72)
20 READ A$, B$
30 INPUT M,N
40 T$ = A$
50 IF B$(M;N) > A$(M;N) THEN DO
60   A$ = B$
70   B$ = T$
80 DOEND
90 IMAGE//"SUBSTRING(",K,"",K,")"
100 PRINT USING 90; M,N
110 PRINT A$(M,N), LIN(1), B$(M,N)
120 PRINT PRINT A$, LIN(1), B$
130 DATA ...
140 DATA ...

04 REM
05 REM INTERCHANGE A$ AND B$ SO THAT
06 REM A$(M,N) > B$(M,N)
```

ALGOL RANGE

CONVERT STATEMENT

CONVERT NUMERIC EXPRESSION IQ STRING VARIABLE

CONVERT STRING EXPRESSION IQ NUMERIC VARIABLE [STATEMENT NUMBER]

1. TRUNCATE THE RESULT, IF STRING VARIABLE IS TOO SHORT.
2. THE STRING EXPRESSION MUST CONTAIN A VALID NUMERIC CONSTANT.

```
10 A = 10, B = 15
20 DIM A1$ [10], A2$ [10], A3$ [10]
30 CONVERT A+B TO A1$
40 CONVERT -25 TO A2$
50 CONVERT 50 TO A3$

60 CONVERT A2$ + A3$ TO C
70 CONVERT A1$ TO C1
80 CONVERT "(75.5, 105.7)" TO C2
```

*RIGHT END
Truncation*

PURPOSE _____

NAME TEC12

```
10  REM
20  REM  ENGINEERING SPECIFICATION
30  REM  PART NUMBER DATA
40  REM  P8(I,J)
50  REM  I, PART NUMBER LLLL;
60  REM

70  REM  P8(I,1), PART NUMBER HIGH (HHHH);
80  REM  P8(I,2), SUBASSEMBLY 1 QUANTITY REQUIRED;
90  REM  P8(I,3), SUBASSEMBLY2;
100 REM  P8(I,4), SUBASSEMBLY3;
110 REM  P8(I,5), SUBASSEMBLY4;
120 REM  P8(I,6), SUBASSEMBLY5;
130 REM  P8(I,7), INVENTORY NUMBER HIGH (HH);
140 REM  P8(I,8), INVENTORY NUMBER LOW (LLLL).
150 REM
```

INST.

NAME

handout

MODULE: _____

TOPIC REF: _____

APPLICABLE REF: _____

PURPOSE _____

INST.

NAME

```
160 REM PART NUMBER RECORD.  
170 REM 21-89.  
180 DIM P8(89,8)  
190 REM  
200 REM PART NUMBER DATA.  
210 DIM C9$ (80)  
  
300 REM  
310 REM GET RECORD  
320 LINPUT C9$  
330 IF LEN (C9$)=0 THEN END  
340 REM CONVERT RECORDNUMBER  
335 REM  
350 CONVERT C9$(5;4) TO I  
360 CONVERT C9$(1;4) TO P8(I,1)  
370 REM  
380 REM SUBASSEMBLY QUANTITY  
390 J = 2  
400 FOR K = 9 TO 24 STEP 3  
410 CONVERT C9$(K;3) TO P8(I,J)  
420 J = J+1  
430 NEXT K  
440 REM  
450 REM CONVERT INVENTORY NUMBER  
460 CONVERT C9$(24;2) TO P8(I,7)  
470 CONVERT C9$(26;4) TO P8(I,8)  
480 REM  
490 REM NEXT RECORD  
500 GOTO 320
```

MODULE: _____

TOPIC REF: _____

APPLICABLE REF: _____

handout

PURPOSE _____

INST.

NAME

```

160 REM PART NUMBER RECORD.
170 REM 21-89.
180 DIM P8(89,8)
190 REM
200 REM PART NUMBER DATA.
210 DIM C9$ (80)

300 REM
310 REM GET RECORD
320 LINPUT C9$
330 IF LEN (C9$)=0 THEN END
340 REM CONVERT RECORDNUMBER
335 REM
350 CONVERT C9$(5;4) TO I
360 CONVERT C9$(1;4) TO P8(I,1)
370 REM
380 REM SUBASSEMBLY QUANTITY
390 J = 2
400 FOR K = 9 TO 24 STEP 3
410 CONVERT C9$(K;3) TO P8(I,J)
420 J = J+1
430 NEXT K
440 REM
450 REM CONVERT INVENTORY NUMBER
460 CONVERT C9$(24;2) TO P8(I,7)
470 CONVERT C9$(26;4) TO P8(I,8)
480 REM
490 REM NEXT RECORD
500 GOTO 320

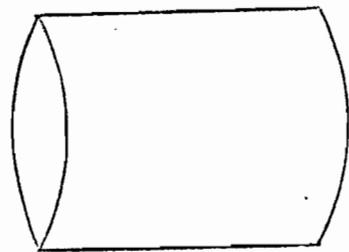
```

BASIC/3000

BINARY FILES

ASCII FILES

BASIC FORMATTED FILES



DASD
FILES



BASIC FORMATTED FILES

>CREATE J1,10

>CREATE J2,10

>CREATE J3,10,300

↑ RECORD SIZE

>PURGE JBAS1

>PURGE JBAS2

FILE IDENTIFIERS

100 FILES J1, J2, J3

200 PRINT #3; "YELLOW TAIL TUNA"

220 RESTORE #1

400 PRINT #2; A,B4,U9

500 MAT READ #1; Z2,B6\$

FILES STATEMENT

100 FILES J1,J2,J3,*,K4

200 FILES M7/MIKE.PUB.STDT, J2.PUB, JT76.PUB.STDT

300 FILES M237, #4,*,*

FILE ACCESS

: FILE JOP2; DEV=TAPE

100 FILES JOP2, QT37, W76

200 PRINT #3; "YELLOW TAIL", END

220 RESTORE #1

400 ON END #2 THEN 9999

500 READ #2; J6, B9\$

600 PRINT Z6, B9\$

700 GOTO 500

9999 END

FILE ACCESS EXAMPLE

10 FILES J1

20 DIM Q(2)

30 REM

40 REM FNT, DATA EDIT ROUTINE.

How many DIMENSIONS

50 IF FNT(Q[*])=0 THEN DO

{ END OF DATA ROUTINE }

230 DOEND

240 GOTO 50

DATA-EDIT ROUTINE

A-23

IF RETURN HAVE VAL FOR DIM 2(1) 7 # VALUE

500 DEF FNT(Z8[*]1) SETS VALUE TO RETURN
510 MAT INPUT Z8

520 IF Z8(1) < 0 THEN RETURN = 0
530 ELSE DO
540 IF Z8(1) >= 1 AND Z8(1) <= 5 OR Z8(1) >= 21 AND Z8(1) <= 89 THEN DO
550 REM
560 REM RECORD NUMBER IS VALID.
570 MAT PRINT # 1; Z8
580 RETURN = 1

MAY HAVE LOCAL VARIABLES SAME AS MAIN PROGRAM

590 DOEND
600 ELSE DO
610 REM
620 REM RECORD NUMBER IS NOT CORRECT.
630 INPUT "SUBASSEMBLY(1-5), PART NO. (21-89)? ", Z8(1), Z8(2)
640 GOTO 520

650 DOEND
660 DOEND
670 FNEED

VARIABLE DIM DEFAULT = 10
STRING DIM DEFAULT = 2

END-OF-DATA ROUTINE

```
50 IF FNT(Q[*]) = 0 DO
60 REM
70 REM END-OF-DATA,
80 PRINT #1; LIN(1)
90 PRINT LIN(2)
100 RESTORE #1
110 PRINT "RECORD", "NUMBER", "QUANTITY"
120 ON END #1 THEN 210
130 R = 0
140 REM
150 REM VERIFY THE FILE DATA.
160 MAT READ #1; Q
170 R = R+1
180 PRINT R, Q(1), Q(2)
190 GOTO 150
200 REM
210 REM EOF
220 END
230 DOEND
240 GOTO 50
```

FORCE PHYSICAL WRITE ON DISC

MULTIPLE ENDS ALLOWED

TYP FUNCTION

*0 = NEXT ITEM
IN DATA LIST*

TYP(FILENUMBER)

- 1 REAL
- 2 STRING
- 3 END-OF-FILE MARK
- 4 END-OF-RECORD MARK
- 5 INTEGER
- 6 LONG
- 7 COMPLEX

I1 I2 I3''''''

10 FILES J2, J3
20 PRINT "TYP(1)=▲"; TYP(1), "TYP(2)=▲"; TYP(2)

ADVANCE *viscous*

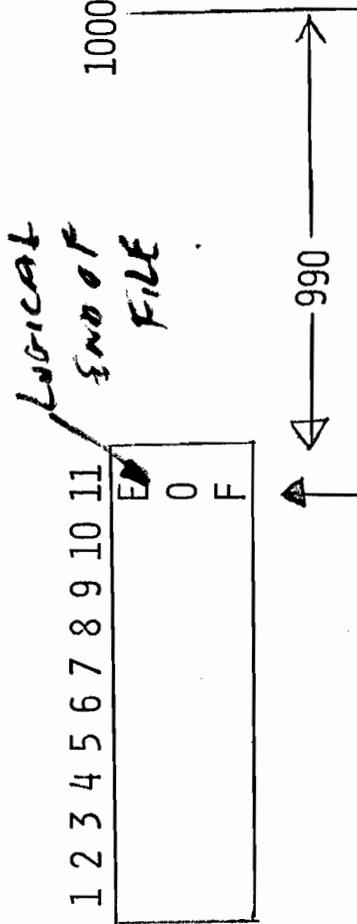
ADVANCE #FILE NUMBER; INTEGER EXPRESSION, TEST VARIABLE

*ITEMS
ADVANCE
FOR*

TEST VARIABLE
0, OK
<>0, EOF

*RETURN {
of items
by ADVANCE
to*

10 FILES J1
20 ADVANCE #1; 1000,T
30 IF T<>0 THEN PRINT "ITEMS=" ; 1000-T



UPDATE

DISC ONLY

IF UPDATE
NUMBER
→

```
100 UPDATE #3; A$
200 RESTORE #2
210 INPUT "UPDATE RECORD, N?"; N
220 ADVANCE #2; N-1, T
230 IF T<>0 THEN DO
240 PRINT "OFF THE END"
250 GOTO 200
260 DOEND
270 INPUT "NEW ITEM, I7?"; I7:
280 UPDATE #2; I7
290 IF BUF(0)=4 THEN 220
300 GO TO 270
```

R/D

DIRECT ACCESS

PRINT # FILE NUMBER, RECORD NUMBER; LIST [END]

READ # FILE NUMBER, RECORD NUMBER; LIST

BASIC FORMATTED FILE

THE FILE IS COMPOSED OF ITEMS WHICH ARE ORGANIZED INTO RECORDS.

THE FILE RESIDES ON A DIRECT ACCESS STORAGE DEVICE (DASD).



- ITEM
- INTEGER VALUE;
 - REAL VALUE;
 - LONG VALUE;
 - COMPLEX VALUE;
 - STRING VALUE;



DASD

7900A CARTRIDGE DISC

Ø	N
---	---

N= 19,488 SECTORS

2888A DISC FILE

Ø

	M
--	---

M= 186,760 SECTORS

128 NOROS

BASIC FORMATTED FILES

System

3000 SECTOR

106 <i>USABLE</i>	<i>words</i> 22
-------------------	--------------------

3000/22 = 136.36
per item

320

ITEMS ...	FORMAT
-----------	--------

RECORD 1

ITEMS	FORMAT
-------	--------

RECORD 2

ITEMS	FORMAT
-------	--------

RECORD N

106 WORDS;
212 WORDS;
319 WORDS.

106 words {
212 words {
319 words {

FILES

PRODUCT NUMBER
1
2
3
4
5

RECORD NUMBER	PRODUCT NUMBER

RECORD

SPECIFICATION

100 PRODUCT RECORDS

FILES

```
10 DIM P1$(8), S(5), P2$(8)
20 FILES SPECIO, REC2
30 INPUT "PRODUCT RECORD?", P1$, (FOR K=1 TO 5, S(K))
40 R1 = REC(1)
50 PRINT #2; R1, P1$
60 FOR I = 1 TO 100
70 PRINT #2; P1$, (FOR K=1 TO 5, S(K))
80 INPUT "PRODUCT RECORDS?", P1$, (FOR K = 1 TO 5, S(K))
90 IF P1$ = "EXIT" THEN END
100 NEXT I
110 PRINT #1; LIN(1)
120 GOTO 30
```

ARRECNO IS CURRENTLY BEING ACCESSED

FILES

```
10 INPUT: " PRODUCT NUMBER?", P1$:
30 IF P1$= "Ø" THEN END
40 REM
50 REM FIND THE RECORD NUMBER
60 GOSUB 800
70 FOR I = 2 TO 100
80 READ #1; P2$, (FOR K = 1 TO 5, S(K))
90 IF P1$= P2$ THEN DO
100 PRINT P1$, LIN(1), (FOR K = 1 TO 5, S(K))
110 GOTO 10
120 DOEND
130 NEXT I
140 PRINT "PRODUCT NUMBER", P1$, "IS NOT HERE"
150 GOTO 10
65 READ #1, R1; P2$, (FOR K = 1 TO 5, S(K))
```

FILES PROGRAM CONTINUED

CREATE TEST VARIABLE, FILE NAME, FILE LENGTH, RECORD SIZE

TEST VARIABLE

- 0, SUCCESSFUL;
- 1, A FILE ALREADY EXISTS WITH THAT NAME;
- 2, THE USER DOES NOT HAVE ENOUGH SPACE LEFT.

} FNEP

PURGE TEST VARIABLE, FILE NAME

TEST VARIABLE

- 0, SUCCESSFUL;
- 1, FILE IS BEING ACCESSED;
- 2, USER IS NOT PERMITTED TO PURGE THIS FILE;
- 3, NO SUCH FILE.

} FNEP


```

10 REM
20 DIM A$(40)
30 INPUT "FILE NAME? ",A$
40 CREATE T, A$, 10, 319
70 IF T=0 THEN DO
80 REM
90 REM ERROR CONDITION
100 IF T=2 THEN DO
110 PRINT "NO SPACE"
120 END
130 DOEND
140 ELSEDO
150 REM
160 REM DUPLICATE FILE NAME.
170 PURGE T1, A$
180 IF T1 = 0 THEN DO
190 PRINT FNE$
200 END
210 DOEND
220 DOEND
230 DOEND

```

print
 ERROR MESSAGE
 STRING

FILES

FILES*

UNASSIGNED FILE

ASSIGN FILE NAME EXPRESSION, FILE NUMBER, TEST VARIABLE
[PROTECT MASK] [RESTRICTION CODE]

{ ASSIGN *, FILE # FILE CLOSE

RESTRICTION CODE, COMMUNITY FILES

RR, READ/WRITE RESTRICTION;
WR, WRITE RESTRICTION, READ-ONLY;
WL, WRITE RESTRICTION WITH DYNAMIC LOCKING;
NR, NO RESTRICTION;
NL, NO RESTRICTION WITH DYNAMIC LOCKING.

RUN
TIME
UNIT

STRING

"

IO SYSTEM, T

MPE

COMMAND

NO COLON

MPE
ERROR
#

ASSIGN

TEST VARIABLE

- 0, THE FILE IS AVAILABLE FOR READ AND WRITE ACCESS;
- 1, THE FILE IS AVAILABLE FOR READ-ONLY ACCESS;
- 3, THE REQUESTED FILE DOES NOT EXIST;
- 5, NO BUFFER SPACE FOR THE FILE;
- 6, THE REQUESTED FILE HAS A RR RESTRICTION;
- 7, USER'S RESTRICTIONS ARE NOT POSSIBLE;
- 8, THE FILE IS A WRITE-ONLY FILE.

ASSIGN STATEMENT

```
10 FILES *,*,*  
20 DIM G$(40), G1$(40), G2$(40)  
30 INPUT "DIRECTORY FILE NAME?", G$  
40 ASSIGN G$, 1, T, WR  
60 INPUT "CODE FILE NAME?", G1$  
70 ASSIGN G1$, 2, T, '145 '205, RR  
90 INPUT "DATA FILE NAM?," G2$  
100 ASSIGN G2$, 2, T, NR  
110 GOSUB 220  
  
200 REM  
210 STOP  
220 IF T<>0 THEN DO  
230 PRINT "ASSIGN TEST= ",T  
240 END  
250 DOEND  
260 RETURN
```

DYNAMIC LOCKING

LOCK # FILE NUMBER

UNLOCK # FILE NUMBER

10 FILES *,*,*
20 ASSIGN H1\$,1,T,WL
30 GO SUB 220
40 ASSIGN H2\$, 3,T, NL
50 GOSUB 220
60 LOCK # 1

FILES

BASIC FILE
BINARY FILE
ASCII FILE

FILES FILE REFERENCE

10 PRINT # 1; A,B,C
20 ON END # 3 THEN 9999
30 READ # 3; A\$
40 MAT READ #2; A,B,C

ADVANCE, UPDATE, CREATE

BASIC FILES ONLY ⇒ DISC

FILES

BINARY FILE

REAL
INTEGER
LONG
COMPLEX

STRING

OUTPUT WRITE 1 BYTE/CHARACTER,
INTEGRAL NUMBER OF WORDS,
INPUT

$\frac{\text{LEN (A\$) + 1}}{2}$

A\$ DIMENSIONED PHYSICAL LENGTH OF A\$

A\$ (I)

A\$ (M,N)

A\$ (M;N)

ASCII

TREATED AS FORMATTED TERMINAL FILES;

LINPUT # FILE NUMBER [RECORD NUMBER] ; STRING VARIABLE

*MANAGE TAPE
READ ACCORDING
TO DIM. output
∴ PAD*

FILES

DASD

NEW FILES
OLD FILES

I/O DEVICE FILES

MAGNETIC TAPE
CARD READER
CARD PUNCH
LINE PRINTER
TERMINAL

OLD FILES

TEMPORARY
PERMANENT

BASIC
SPL
COBOL
FORTRAN
:BUILD



FILES

: FILE FORMAL DESIGNATOR

1. EQUATE FILE NAMES;
2. SPECIFY FILE CHARACTERISTICS;
3. MODIFY FILE CHARACTERISTICS.

SYSTEM TEST VARIABLE, STRING EXPRESSION

```
: FILE TO; DEV = TAPE; REC = -2400; NOLABEL
: BUILD SCRATCH; REC = 256; DISC = 2000, 16, 16
: FILE TI; DEV = TAPE; REC = 1037,, V; NOBUF
: FILE TERMI; REC = - 72,, , ASCII; DEV = TAPE
: FILE SPOOL, NEW; REC = 5L2,, V, ASCII; SAVE
: PTAPE SPOOL
```

ESP.
IBM

→ 2000c To
3000

LOOK EXACTLY
THE PRINTOUT
YOU WANT

SYSTEM COMMANDS

EXIT
KEY
SPOOL
TAPE
XEQ ASCIIFILE [, ECHO]

→

LINE
TYPE
ON TAPE
FILE
FILE

↑
BUILD VIA TEXT EDITOR

BASIC FILE TYPE OF BINARY FILE
IS ANOTHER

GET T1
LIST, OUT=PRINTER

NOES → ^{SEGMENTATION} ~~PROGRAM NAME~~
END STOP - TERMINATES SEQUENCE

— ALL FILES CLOSED

CHAIN NAME EXPRESSION , LABEL EXPRESSION

— FILE STILL OPEN

INVOKE NAME EXPRESSION , LABEL EXPRESSION

CALL P (PARAMETER LIST)

* P (PARAMETER LIST)

} GET NON-BASIC PROCEDURES

COM

COM (1)

COM (2)

.

.

COM (9)

MULTIPLE COM BLOCKS

∴ NOT ALL PROGRAMS

NEED ALL IN MEM. STATEMENTS

FILES # INTEGER

REOPEN A FILE WITHIN INVOKE S.O.

EXTERNAL
GOSUB

BASIC 1.00

20 REM: MAY 12, 1972

30 REM: 5

100 REM: ALGORITHM E. (DATE OF EASTER). LET Y BE THE YEAR FOR WHICH THE DATE OF EASTER IS DESIRED.

120 REM:

107 REM:
110 REM: E1. (GOLDEN NUMBER.) SET G← (YMOD19)+1. (G IS THE SO-CALLED "GOLDEN NUMBER" OF THE YEAR IN THE 19-YEAR METONIC CYCLE.)

115 REM:

120 REM:
125 REM:
130 REM: E2. (CENTURY.) SET C← FLOOR(Y/100)+1. (WHEN Y IS NOT A MULTIPLE OF 100, C IS THE CENTURY NUMBER; I.E., 1970 IS IN THE TWENTIETH CENTURY.)

135 REM:

140 REM:
145 REM:
150 REM: E3. (CORRECTIONS.) SET X1← FLOOR(3*C/4)-12, Z← FLOOR((8*C+5)/25)-5. (X1 IS THE NUMBER OF YEARS, SUCH AS 1900, IN WHICH LEAP YEAR WAS DROPPED IN ORDER TO KEEP IN STEP WITH THE SUN. Z IS A SPECIAL CORRECTION DESIGNED TO SYNCHRONIZE EASTER WITH THE MOON'S ORBIT.)

155 REM:

160 REM: E4. (FIND SUNDAY.) SET D← FLOOR(5*Y/4)-X1-10. (MARCH((-D)MOD7) ACTUALLY WILL BE A SUNDAY.)

165 REM:

170 REM: E5. (EPOCH.) SET E← (11*G+20+Z-X1) MOD 30. (IF E=25 AND THE GOLDEN NUMBER G IS GREATER THAN 11, OR IF E=24, THEN INCREASE E BY 1. (E IS THE SO-CALLED "EPOCH", WHICH SPECIFIES WHEN A FULL MOON OCCURS.)

175 REM:

180 REM: E6. (FIND FULL MOON.) SET N← 44-E. IF N<21 THEN SET N← N+30. (EASTER IS SUPPOSEDLY THE "FIRST SUNDAY FOLLOWING THE FIRST FULL MOON WHICH OCCURS ON OR AFTER MARCH 21.")

185 REM:

190 REM: (ACTUALLY PERTURBATIONS IN THE MOON'S ORBIT DO NOT MAKE THIS STRICTLY TRUE, BUT WE ARE CONCERNED HERE WITH THE "CALLNDAR MOON" RATHER THAN THE ACTUAL MOON. THE NTH OF MARCH IS A CALENDAR FULL MOON.)

195 REM:

200 REM: E7. (ADVANCE TO SUNDAY.) SET N← N+7-((D+N) MOD 7).

205 REM:

210 REM: E8. (GET MONTH.) IF N>31, THE DATE IS (N-31) APRIL; OTHERWISE THE DATE IS N MARCH.

215 REM:

410 REM:

111

```
4 REM
5 REM
10 REM VOLUME OF A SPHERE,  $V=4/3$  3
20 REM THE AVERAGE RADIUS OF THE EARTH IS 6.371 KILOMETERS
30 REM THE AVERAGE RADIUS OF THE MOON IS 1.738 KILOMETERS
36 REM THE VOLUME OF THE EARTH IS  $1.082 \times 10^{21}$  CUBIC METERS
40 REM  $V(E)/V(M)$ 
45 REM V IS THE VOLUME OF THE MOON
50  $A=6371^3/1738^3$ 
55 PRINT " V(E)/V(M)=";A
60  $V=1/A^*1.082E24$ 
70 PRINT "V(M)=";V
```

110

```
10 GOTO 40
20 PEM C5, COMMAND; A1, DATA; A2, RESULT-
30 PRINT A2
40 READ C5, A1
50 GOTO C5 OF 70, 120, 170, 200
60 END
70 REM
80 REM A2 MINUTES PER KILOMETER; A1 KILOMETERS PER HOUR.
90 A2=60/A1
100 GOTO 30
110 REM
120 REM A2, MILES PER HOUR; A1, MINUTES PER KILOMETER.
130 REM 5/6 MILES PER KILOMETER.
140 A2=50/A1
150 GOTO 30
160 REM
170 REM A2, FEET PERSECOND; A1 MILES PER HOUR.
180 A2=A1*5280*60**-2
190 GOTO 30
200 REM
210 REM A2, CENTIMETERS PER SECOND; A1, MINUTES PER MILE.
220 A2=1/A1*6/5*1000/60
230 GOTO 30
```

T7

```
5 REM A0, DATA.
10 REM A1 SUBTOTAL.
20 REM A2, TOTAL.
30 REM A2$, CONTROL CHARACTER.
40 A1=A2=:
50 INPUT :
60 IF BUF(0)=2 THEN DO
70   INPUT :A2$:
80   PRINT "SUBTOTAL ";A2$,LIN(1),A1
90   A2=A2+A1,A1=0
100 DOEND
110 IF BUF(3)=4 THEN DO
120   REM FINI
130   REM TOTAL LINE
140   PRINT "TOTAL=↑";A2+A1
150   END
160 DOEND
170 REM
180 REM NUMERIC DATA
190 INPUT :A0:
200 A1=A1+A0
210 GOTO 60
220 REM TEST THE NEXT ITEM
```


T9

```
10 REM COMPOUND INTEREST.
20 REM S, COMPOUND SUM.
30 REM P, AMOUNT INVESTED.
40 REM N, NUMBER OF YEARS INVESTED.
50 REM I, INTEREST
60 P=1
70 PRINT .01,.02,.03,.04,.05
80 DIM S(5)
90 FOR N=1 TO 15
100   FOR I=.01 TO .05 STEP .01
110     II=I*100
120     S[II]=P*(I+I)**N
130   NEXT I
140   PRINT (FOR J=1 TO 5,S[II])
150 NEXT N
```

```
5 REM
10 REM C. KNUTH VOLUME 3.
20 REM STRAIGHT INSERTION SORT.
30 REM S(16) DATA ELEMENTS.
40 REM X, TEMPORARY.
50 REM I, MAJOR INDEX
60 REM J, MINOR INDEX
65 DIM S(16)
68 N=16
70 READ (FOR K=1 TO N,S(K))
75 PRINT (FOR K=1 TO N,K;S(K),LIN(1))
90 REM
100 FOR I=2 TO N
110   X=S(I)
120   FOR J=I-1 TO 1 STEP -1
130     IF X>=S(J) THEN 160
140     S(J+1)=S(J)
150   NEXT J
160   S(J+1)=X
170 NEXT I
180 PRINT (FOR I=1 TO 16,S(I),LIN(1))
190 REM
200 DATA 543,87,512,61,908,170,897,275
210 DATA 653,426,154,509,612,677,765,703
```

T5

```
10 REM SUM OF THE YEARS DIGITS DEPRECIATION
20 REM S, SUM OF THE DIGITS
30 REM N, ASSET'S LIFE IN YEARS
40 READ N
47 IF N=2 THEN END
50 REM
60 REM SUM OF THE YEARS DIGITS
70 REM IF N=3, SET S:=6 1+2+3=6.
72 REM N=3,
74 REM THE RATE IS 3/6; 2/6, 1/6
75 S=0
80 FOR I=1 TO N
90   S=S+I
100 NEXT I
110 PRINT "YEAR","RATE"
120 REM
130 REM PRINT A TABLE OF DEPRECIATION RATES
135 R1=0
140 FOR I=0 TO N-1
145   R=(N-I)/S
147   R1=R+R1
150   PRINT I+1,R
160 NEXT I
162 PRINT "R1= ",R1,LIN(1)
165 GOTO 45
170 DATA 8,16,0
```

```
10 REM
20 REM INTERCHANGE A$ AND B$ SO THAT
30 REM A$(M,N)>B$(M,N)
40 DIM A$(72),B$(72),T$(72)
50 DIM A1$(72)
60 FOR I=1 TO 61 STEP 10
70   A1$(I:10)="1234567390"
80 NEXT I
90 PRINT "INPUT LINE A$ AND LINE B$?"
100 PRINT A$
110 INPUT A$
120 INPUT B$
130 IF LEN(A$)=0 OR LEN(B$)=0 THEN END
140 INPUT "M,N ",M,N
150 IF M<0 OR N<0 THEN 90
160 T$=A$
170 IF B$(M:N)<A$(M:N) THEN DO
180   A$=B$
190   B$=T$
200 DOEND
210 IMAGE 7/"SUBSTRING(",K,"",K,")"
220 PRINT USING 210;M,N
230 PRINT A$(M,N),LIN(1),B$(M,N)
240 PRINT A$,LIN(1),B$
250 GOTO 140
```

```
5  PEM CIPHER - CODES AND DECODES MESSAGES
10 DIM A$(72),B$(72),C$(72),C1$(72),B1$(72)
20 K3=3
30 PRINT
40 PRINT "YOUR CLEARTEXT MESSAGE IS?"
50 INPUT B$
55 A1=LEN(B$)
58 IF A1=0 THEN END
60 PRINT B$
70 PRINT
80 PRINT "YOUR CIPHERTEXT MESSAGE IS?"
100 FOR I=1 TO A1
130  C$(I)=CHR$(NUM(B$(I))-3)
140 NEXT I
150 PRINT C$
155 PRINT "CIPHERTEXT MESSAG IS?"
160 INPUT C1$
170 A1=LEN(C1$)
175 IF A1=0 THEN END
180 FOR I=1 TO A1
190  B1$(I)=CHR$(NUM(C1$(I))+3)
200 NEXT I
210 PRINT LIN(1)," YOUR CLEARTEXT IS "
220 PRINT B1$
230 GOTO 30
```

T2

```
5 REM
10 REM CONVERT HEX DIGITS TO DECIMAL.
20 REM TERMINATE ON AN INVALID CHARACTER.
25 REM (1,2 = 15), (2,1 = 33), (1,A = 26), ("A","a","F" = 2751)
30 REM A1, DATA
40 REM A0, DECIMAL NUMBER.
45 A0=0
50 INPUT :
52 REM
54 REM HEX CHARACTER
56 REM A,B,C,D,E,F
60 IF BUF(0)=2 THEN DO
70 INPUT :A1:
80 IF A1<"A" OR A1>"F" THEN END
90 A1=NUM(A1)-55
100 GOTO 210
110 DOEND
111 REM
112 REM IF THE BUFFER IS EMPTY
113 REM PRINT THE RESULT
114 REM AND GET THE NEXT VALUE.
120 IF BUF(0)=4 THEN DO
130 PRINT A0
135 PRINT LIN(1)
140 A0=0
150 GOTO 50
160 DOEND
170 IF BUF(0)<>5 THEN END
180 REM
190 REM NEXT ITEM IS AN INTEGER HEX DIGIT
200 INPUT :A1:
210 A0=A0*16+A1
220 GOTO 60
```

114

```
10 FILES 199
20 ON END #1 THEN 400
30 DIM A$(40)
35 AV=0
40 READ #1;A$
50 AV=AV+1
60 PRINT AV;A$
70 GOTO 40
400 END
```

T)

```
10 REM
20 REM ENGINEERING SPECIFICATION
30 REM PART NUMBER DATA
40 REM P8(I,J)
50 REM 1. PART NUMBER LLLL;
60 REM
70 REM P8(I,1), PART NUMBER HIGH (HHHH);
80 REM P8(I,2), SUBASSEMBLY 1 QUANTITY REQUIRED;
90 REM P8(I,3), SUBASSEMBLY 2;
100 REM P8(I,4), SUBASSEMBLY 3;
110 REM P8(I,5), SUBASSEMBLY 4;
120 REM P8(I,6), SUBASSEMBLY 5;
130 REM P8(I,7), INVENTORY NUMBER HIGH (HH),
140 REM P8(I,8), INVENTORY NUMBER LOW (LLLL).
150 REM
160 REM PART NUMBER RECORD.
170 REM 21-89.
180 INTEGER P8[89,8]
185 MAT P8=CON
190 REM
200 REM PART NUMBER DATA.
210 DIM C9$(80)
300 REM
305 PRINT "PARTNUMBER RECORD? "
310 REM GET RECORD
320 LINPUT C9$
330 IF LEN(C9$)=0 THEN DO
332 PRINT (FOR I=1 TO 89,LIN(2), (FOR J=1 TO 8,P8[I,J],LIN(1)))
334 END
335 REM
336 DOEND
340 REM CONVERT RECORDNUMBER
350 CONVERT C9$(5:4) TO I
360 CONVERT C9$(1:4) TO P8[I,1]
370 REM
380 REM SUBASSEMBLY QUANTITY
390 J=2
400 FOR K=9 TO 24 STEP 3
410 CONVERT C9$(K:3) TO P8[I,J]
420 J=J+1
430 NEXT K
440 REM
450 REM CONVERT INVENTORY NUMBER
460 CONVERT C9$(24:2) TO P8[I,7]
470 CONVERT C9$(26:4) TO P8[I,8]
480 REM
490 REM NEXT RECORD
495 PRINT (FOR I=1 TO 8,P8[21,I],LIN(1))
500 GOTO 320
```



```
1 GET T1
2 LIST ,OUT=PRINTER
3 GET T2
4 LIST ,OUT=PRINTER
5 GET T3
6 LIST ,OUT=PRINTER
7 GET T4
8 LIST ,OUT=PRINTER
9 GET T5
10 LIST ,OUT=PRINTER
11 GET T6
12 LIST ,OUT=PRINTER
13 GET T7
14 LIST ,OUT=PRINTER
15 GET T8
16 LIST ,OUT=PRINTER
17 GET T9
18 LIST ,OUT=PRINTER
19 GET T10
20 LIST ,OUT=PRINTER
21 GET T11
22 LIST ,OUT=PRINTER
23 LIST ,OUT=PRINTER
24 GET T12
25 LIST ,OUT=PRINTER
```


:FORTRAN [sourcefile] [, [uslfile] [, [listfile]
[, [masterfile] [, newfile]]]]

:FREERIN rin

:GETRIN [rinpassword]

:HELLO [sessionname.] username [/upass]
.acctname [/apass] [, group [/gpass]]
[;TERM=termtype] - 3 export
[;PRI=executionpriority]

:JOB [jobname.] username [/upass]
.acctname [/apass] [, groupname
[/gpass]]
[;TERM=termtype]
[;PRI=executionpriority]
[;OUTCLASS=outputclass]

:LISTF [filesset] [, detail] [, listfile]

:PREP uslfile, progfile
[;ZERODB]
[;PMAP]
[;MAXDATA=segsize]
[;STACK=stacksize]
[;DL=dsize]
[;CAP=caplist]
[;RL=filename]

:PTAPE [filename]

:PURGE filereference [, TEMP]

:RENAME *oldfilereference,newfilereference*
 [*,TEMP*]

:RESTORE *storedset* [;*filesetlist*]
 [;*dev*] [;*KEEP*]]

:RESUME

:RESET *formaldesignator*

:RUN *progfile* [*,entrypoint*]
 [;*NOPRIV*]
 [;*LMAP*]

 [;*MAXDATA=segsz*]
 [;*PARM=parameternum*]
 [;*STACK=stacksize*]
 [;*DL=dlsz*]
 [;*LIB=library*]

:SAVE { [*\$OLDPASS,newfilereference*]
 { [*tempfilereference*]

:SEGMENTER [*listfile*]

:SHOWTIME

:SPEED [*inspeed*] [*,outspeed*]

:SPL [*sourcefile*] [,*uslfile*]
 [,*listfile*] [,*masterfile*]
 [,*newfile*]]]

:STAR [*listfile*] [,*NOLIST*]

:STORE [*filesetlist*] *destfile*

:TELLOP *message*

C.I. ERRORS FORMAT: ERR XXX[Y]

WHERE XXX=ERROR NUMBER

Y =PARAMETER WHICH CAUSED ERROR (OPTIONAL)

- 1 UNKNOWN COMMAND
- 2 ABNORMAL PROGRAM TERMINATION
- 3 NOT ALLOWED FROM SESSIONS
- 4 CAPABILITY CONFLICT
- 5 TOO MANY PARAMETERS
- 6 INSUFFICIENT PARAMETERS
- 7 MISSING COLON
- 8 ILLEGITIMATE ACCESS
- 9 UNACCEPTABLE DEVICE
- 10 INSUFFICIENT RESOURCES
- 11 COMMAND NOT YET IMPLEMENTED
- 12 NOT ALLOWED FROM A JOB
- 13 VDD FULL
- 14 JOB OVERLOAD
- 15 SUBSYSTEM NOT FOUND
- 16 SYNTAX ERROR
- 17 PARAMETER NOT OPTIONAL
- 18 ILLEGAL PARAMETER
- 19 PARAMETER OUT OF BOUNDS
- 20 SYNTAX ERROR
- 21 PARAMETER NOT OPTIONAL
- 22 ILLEGAL PARAMETER
- 23 PARAMETER OUT OF BOUNDS
- 24 ILLEGAL PARAMETER IN THIS CONTEXT
- 25 DUPLICATE PARAMETER
- 26 ILLEGAL KEYWORD
- 27 DUPLICATE KEYWORD
- 28 ILLEGAL KEYWORD IN THIS CONTEXT
- 29 ILLEGAL NAME
- 30 INVALID NUMBER
- 31 INVALID NUMBER OF BUFFERS
- 32 ILLEGAL ACCESS TYPE
- 33 ILLEGAL NUMBER OF EXTENTS
- 34 ILLEGAL INITIAL ALLOCATION VALUE
- 35 ILLEGAL BLOCKING FACTOR
- 36 ILLEGAL FILE NAME
- 37 FILE EQUATION TABLE FULL
- 38 BACK FILE REFERENCE NOT FOUND
- 39 TOO MANY BACK FILE REFERENCES
- 40 INVALID FILE DESIGNATOR
- 41 ILLEGAL DEVICE NAME
- 42 ILLEGAL CAPABILITY
- 43 DUPLICATE CAPABILITY
- 44 ILLEGAL LIBRARY SPECIFIED
- 45 UNABLE TO ACCESS TERMINAL FILE
- 46 INSUFFICIENT SYSTEM RESOURCES
- 47 ILLEGAL DEVICE OR INVALID INPUT SPEED
- 48 ILLEGAL DEVICE OR INVALID OUTPUT SPEED
- 49 SURQUE IS LINEAR AND HAS NO QUANTUM
- 50 INVALID NUMBER OF BUFFERS
- 51 ILLEGAL ACCESS TYPE

52 ILLEGAL NUMBER OF EXTENTS
53 ILLEGAL INITIAL ALLOCATION VALUE
54 ILLEGAL BLOCKING FACTOR
55 ILLEGAL FILE NAME
FILE EQUATION TABLE FULL
BACK FILE REFERENCE NOT FOUND
58 TOO MANY BACK FILE REFERENCES
59 INVALID FILE DESIGNATOR
60 ILLEGAL DEVICE NAME
61 ILLEGAL CAPABILITY
62 DUPLICATE CAPABILITY
63 ILLEGAL LIBRARY SPECIFIED
64 UNABLE TO ACCESS TERMINAL FILE
65 INSUFFICIENT SYSTEM RESOURCES
66 ILLEGAL DEVICE OR INVALID INPUT SPEED
67 ILLEGAL DEVICE OR INVALID OUTPUT SPEED
68 SUBQUE IS LINEAR AND HAS NO QUANTUM
69 UNKNOWN SUBQUE
70 RIN CURRENTLY IN USE
71 RIN NOT ALLOCATED TO THIS USER
72 RIN TABLE FULL
73 DIRECTORY ERROR
74 INVALID LIST FILE
75 UNDEFINED JOB NAME
76 MESSAGE ROUTING ERROR
77 STORE ERROR
78 RESTORE ERROR
79 NON-EXISTENT USER
80 FILE SYSTEM ERROR
81 UNIT NOT READY
NO WRITE RING
83 INCONSISTENT FILE OPERATION
84 PRIVILEGED FILE VIOLATION
85 INSUFFICIENT DISC SPACE
86 NON-EXISTENT ACCOUNT
87 NON-EXISTENT GROUP
88 NON-EXISTENT FILE
89 INVALID FILE NAME
90 DEVICE UNAVAILABLE
91 INVALID DEVICE SPECIFICATION
92 NO PASSED FILE
93 EXCLUSIVE VIOLATION
94 LOCKWORD VIOLATION
95 SECURITY VIOLATION
96 DUPLICATE NAME
97 DIRECTORY OVERFLOW
98 ATTEMPT TO SAVE SYSTEM FILE AS JOB TEMPORARY
99 IN USE: CAN'T BE PURGED
100 FILE SYSTEM ERROR
101 UNIT NOT READY
102 NO WRITE RING
103 INCONSISTENT FILE OPERATION
104 PRIVILEGED FILE VIOLATION
105 INSUFFICIENT DISC SPACE
NON-EXISTENT ACCOUNT
NON-EXISTENT GROUP
108 NON-EXISTENT FILE
109 INVALID FILE NAME
110 DEVICE UNAVAILABLE
111 INVALID DEVICE SPECIFICATION

112 NO PASSED FILE
113 EXCLUSIVE VIOLATION
114 LOCKWORD VIOLATION
115 SECURITY VIOLATION
116 DUPLICATE NAME
117 DIRECTORY OVERFLOW
118 ATTEMPT TO SAVE SYSTEM FILE AS JOB TEMPORARY
119 IN USE: CAN'T BE PURGED
120 CREATE ERROR
121 LOAD ERROR
122 ILLEGAL LIBRARY SEARCH
123 UNKNOWN ENTRY POINT
124 DATA SEGMENT TOO LARGE
125 STACK SIZE TOO SMALL
126 DATA SEGMENT LARGER THAN MAXDATA SPECIFICATION
127 ILLEGAL NUMBER OF CODE SEGMENTS
128 INVALID PROGRAM FILE
129 CODE SEGMENT TOO LARGE
130 MORE THAN ONE EXTENT IN PROGRAM
131 TOO MANY PROCEDURES LOADED
132 UNKNOWN PROCEDURE NAME
133 INVALID PROCEDURE NUMBER
134 SYSTEM SL ACCESS ERROR
135 PUBLIC SL ACCESS ERROR
136 GROUP SL ACCESS ERROR
137 PROGRAM FILE ACCESS ERROR
138 LIST FILE ACCESS ERROR
139 INVALID SYSTEM SL FILE
140 INVALID PUBLIC SL FILE
141 INVALID GROUP SL FILE
142 INVALID LIST FILE
143 ILLEGAL PROCEDURE DEALLOCATION
144 ILLEGAL PROGRAM ALLOCATION
145 ILLEGAL PROCEDURE DEALLOCATION
146 ILLEGAL PROCEDURE ALLOCATION
147 ILLEGAL CAPABILITY
148 UNABLE TO OBTAIN CST ENTRIES
149 UNABLE TO OBTAIN PROCESS DST ENTRY
150 UNABLE TO OBTAIN VIRTUAL MEMORY
151 TRACE SUBSYSTEM NOT PRESENT
152 PROGRAM LOADED IN OPPOSITE MODE
153 SEGMENTER ERROR
154 MISC. I/O ERROR
155 DEFAULT VALUES TAKEN
156 MESSAGE TOO LONG
157 USER NOT ACCEPTING MESSAGES
158 PROGRAM ALREADY ALLOCATED
159 PROCEDURE ALREADY ALLOCATED
160 PROGRAM NOT ALLOCATED
161 PROCEDURE NOT ALLOCATED
162
163
164
165
166
167
168
169
170
171

172
173 LDEV# NO WRITE RING

174

175

178

179

180

181

182

183

184

185

186

187

188

189

190

191

192

193

194

195

196

197

198

199

CREATE ERROR

LOAD ERROR

2 ILLEGAL LIBRARY SEARCH

203 UNKNOWN ENTRY POINT

204 DATA SEGMENT TOO LARGE

205 STACK SIZE TOO SMALL

206 DATA SEGMENT LARGER THAN MAXDATA SPECIFICATION

207 ILLEGAL NUMBER OF CODE SEGMENTS

208 INVALID PROGRAM FILE

209 CODE SEGMENT TOO LARGE

210 MORE THAN ONE EXTENT IN PROGRAM

211 TOO MANY PROCEDURES LOADED

212 UNKNOWN PROCEDURE NAME

213 INVALID PROCEDURE NUMBER

214 SYSTEM SL ACCESS ERROR

215 PUBLIC SL ACCESS ERROR

216 GROUP SL ACCESS ERROR

217 PROGRAM FILE ACCESS ERROR

218 LIST FILE ACCESS ERROR

219 INVALID SYSTEM SL FILE

220 INVALID PUBLIC SL FILE

221 INVALID GROUP SL FILE

222 INVALID LIST FILE

223 ILLEGAL PROCEDURE DEALLOCATION

224 ILLEGAL PROGRAM ALLOCATION

225 ILLEGAL PROCEDURE DEALLOCATION

226 ILLEGAL PROCEDURE ALLOCATION

227 ILLEGAL CAPABILITY

228 UNABLE TO OBTAIN CST ENTRIES

229 UNABLE TO OBTAIN PROCESS DST ENTRY

230 UNABLE TO OBTAIN VIRTUAL MEMORY

231 TRACE SUBSYSTEM NOT PRESENT

232 PROGRAM LOADED IN OPPOSITE MODE
233 SEGMENTER ERROR
234 MISC. I/O ERROR
235 DEFAULT VALUES TAKEN
MESSAGE TOO LONG
USFR NOT ACCEPTING MESSAGES
238 PROGRAM ALREADY ALLOCATED
239 PROCEDURE ALREADY ALLOCATED
240 PROGRAM NOT ALLOCATED
241 PROCEDURE NOT ALLOCATED
242
243
244
245
246
247
248
249
250 SEGMENTER ERROR
251 MISC. I/O ERROR
252 DEFAULT VALUES TAKEN
253 MESSAGE TOO LONG
254 USER NOT ACCEPTING MESSAGES
255 PROGRAM ALREADY ALLOCATED
256 PROCEDURE ALREADY ALLOCATED
257 PROGRAM NOT ALLOCATED
258 PROCEDURE NOT ALLOCATED
259
260
261
262
263
264
265
266
267
268
269
270 LDEV# NO WRITE RING
271
272
273
274
275
276
277
278
279
280

CIERROR

```
10 FILFS L
20 INTFGER N,I[10],B[12]
30 DIM A$(80)
40 MAT READ B
50 PRINT #1;&
*****COMMAND INTERPRETER ERRORS*****
46 PRINT #1
50 DATA 0,46,20,62,50,73,100,103,200,123,250,156
60 PRINT #1;"C.1. ERRORS FORMAT: ERR XXX, Y"
70 PRINT #1
80 PRINT #1;" WHERE XXX=ERROR NUMBER"
90 PRINT #1;" Y =PARAMETER WHICH CAUSED ERROR"
100 PRINT #1;LIN(2)
110 FOR K=1 TO 280
120 FOR L=11 TO 0 STEP -2
130 IF K>=B[L] THEN 150
140 NEXT L
150 N=B[L+1]+K-B[L]
160 #MESGET(N,I[*],A$[*])
170 IF N=-1 THEN PRINT #1;K
180 ELSE PRINT #1;K;A$
190 NEXT K
200 END
```