WANG BASIC LANGUAGE SUMMARY (2200T, PCS-II, PCS-IIA, WS, WCS-15)

BASIC STATEMENTS

ADD [C] $\left(\text{alpha variable, } \begin{Bmatrix}\text{hh}\\ \text{alpha variable}\end{Bmatrix}\right)$

Example: 100 ADDC (A$, 30)

Binary addition, with or without carry, of the two arguments.

AND $\left(\text{alpha variable, } \begin{Bmatrix}\text{hh}\\ \text{alpha variable}\end{Bmatrix}\right)$

Example: 100 AND (A$, B$)

'AND' the second argument into the first.

BIN (alpha variable) = expression

Example: 100 BIN (A$) = X + 4

Convert the integer value of the expression to a one-byte binary value.

BOOL h $\left(\text{alpha variable, } \begin{Bmatrix}\text{hh}\\ \text{alpha variable}\end{Bmatrix}\right)$

(h = hexdigit 0 to 9, A to F)

Example: 100 BOOLE (A$, 3C)

Perform the Boolean operation specified, on the arguments given.

COM com element [, com element] .

Example: 10 COM X, Y$, A(3)

Reserve space for specified common variables.

COM CLEAR $\begin{Bmatrix}\text{scalar variable}\\ \text{array designator}\end{Bmatrix}$

Example: 10 COM CLEAR X

Change common variables to non-common or vice-versa.

CONVERT expression TO alpha variable, (image)

Example: 100 CONVERT X TO A$, (##)

Convert the expression to an alpha variable according to the specified image.

CONVERT alpha variable TO numeric variable

Example: 100 CONVERT A$ TO X

Convert an ASCII-character number to a numeric value.

DATA d [, d] .

Example: 100 DATA"ABC", 3.41

Provide data constants for a READ statement.

DEFFN a(v) = expression

Example: 100 DEFFN F(X) = A*EXP(X)

Define a mathematical function of the variable v with an identifier a.

DEFFN' integer (variable [, var] . .)

Example: 100 DEFFN'0 (A, C$, B(3))

Define a special function key or subroutine entry point with argument passing capability.

WANG
LABORATORIES, INC.
ONE INDUSTRIAL AVE., LOWELL, MA. 01851 (617) 459-5000

```
DEFFN' integer ⎡"character string"      ⎡;   "character string"    ⎤    ⎤
               ⎣HEX (hh[hh]   . . )     ⎣'   HEX (hh [hh] . . . )  ⎦ . .⎦
```

Example: 100 DEFFN' 31 "PRINT"; HEX (220D)

Define a character string to be provided when a special function key is pressed.

DIM dim element [, dim element] . .

Example: 100 DIM A (3,2), B$(5)8

Reserve storage space in memory for the array or alpha variable specified.

END

Example: 900 END

Display free space available in memory; stop execution.

FOR v = expression TO expression [STEP expression]

Example: 100 FOR K = 1 TO 10 + 2

Specify the start of a repetitive loop.

GOSUB line no.

Example: 100 GOSUB 500

Transfer execution to the subroutine at the specified line.

```
GOSUB' integer ⎡⎛"character string"   ⎡   "character string"⎤    ⎞⎤
               ⎢⎜variable             ⎢,  variable          ⎥  . ⎟⎥
               ⎣⎝expression           ⎣   expression        ⎦    ⎠⎦
```

Example: 100 GOSUB' 12 ("JOHN", 12.4,X)

Transfer execution to the subroutine specified.

GOTO line no.

Example: 100 GOTO 200

Transfer execution to the specified line number.

```
HEXPRINT {alpha variable    } [{,} {alpha variable    }]  .    [;]
         {alpha array desig.} [{;} {alpha array desig.}]
```

Example: 100 HEXPRINT A$( ), B$

Print the characters of the variable or array in hexadecimal form.

```
                                   ⎧<  ⎫
                                   ⎪< =⎪
   ⎧literal                      ⎫ ⎪=  ⎪ ⎧literal                      ⎫
IF ⎨alpha or numeric variable    ⎬ ⎨> =⎬ ⎨alpha or numeric variable    ⎬   THEN line no.
   ⎩expression                   ⎭ ⎪>  ⎪ ⎩expression                   ⎭
                                   ⎩< >⎭
```

Examples: 100 IF A < B THEN 40
100 IF STR(A$, I, 2) < HEX(8082) THEN 500

Branch to specified line number if the relation is true.

```
% ' [character string]
  ; [format spec  ;  ]     .
```

Example: 100 % CODE ### ####

Provide the formatting image line for a PRINTUSING statement.

```
     ⎛hh           ⎞ {alpha variable    } [   {alpha variable    }]
INIT ⎜"character"  ⎟ {alpha array desig.} [,  {alpha array desig.}]  .
     ⎝alpha variable⎠
```

Example: 100 INIT (00) A$, B$( ), C$

Initialize the arguments to the character specified.

INPUT ["character string",] variable [, variable] .

Example: 100 INPUT A$, X

Accept values from the keyboard, or other input device.

KEYIN alpha variable, line no. 1, line no. 2

Example: 100 KEYIN K$, 120, 100

Read character if device ready; if normal char., go to line no. 1; if from SF key, go to line no. 2.

[LET] numeric variable [, numeric variable] . . . = expression

Example: 100 A, B = 2 + 4

```
[LET] alpha variable [, alpha variable]  . . = {alpha variable}
                                               {literal       }
```

Example: 100 C$, D$(2) = "ABCDE"

Assign value from right side of equation to variables on left.

NEXT numeric scalar variable

Example: 200 NEXT K

Specify the end of a FOR/NEXT loop.

```
ON expression {GOSUB}  line no. [, line no.]  .
              {GOTO }
```

Example: 100 ON X + 2 GOTO 30, 40, 50, 60

Branch to the ith line no. (i = integer value of expression).

ON ERROR alpha variable, alpha variable GOTO line no.

Example: 100 ON ERROR E$, N$ GOTO 50

Branch to line no. on execution error; put ERR code in first variable, error's line no. in second.

```
OR (alpha variable,  {hh            })
                     {alpha variable})
```

Example: 100 OR (A$, B$)

OR the second argument into the first.

```
PACK (image)  {alpha variable    }  FROM {expression            [, {expression          }]. . .}
              {alpha array desig.}       {numeric array desig.  [  {numeric array desig.}]     }
```

Example: 100 PACK (##.#) A$ FROM X, A( ), M

Pack numeric variables into an alpha variable according to the image.

PRINT [t] [print element] [t print element] . [t] . . . [t] .

Example: 100 PRINT "X=" ; X, T

Output the values specified on a printing or display unit in zoned (t=,) or packed (t=;) format.

PRINTUSING line no. [, print elt t] . . [;]

Example: 100 PRINTUSING 101, A$, T(1), Q

Output values for variables in format of image (line no.) statement; t = , or ;

READ variable [, variable] . .

Example: 100 READ X$, B(3), I, A$(1)

Assign values from DATA statement to the variables specified in the READ statement.

REM [text string]

Example: 100 REM THIS SUBROUTINE PERFORMS I/O OPERATIONS

Provide non-executable explanatory remarks in program text.

RESTORE [expression]

Example: 100 RESTORE X + 7

Return the DATA statement pointer to the ith DATA value. i = truncated integer value of expression.

RETURN

Example: 100 RETURN

Mark end of subroutine; return processing to statement following last executed GOSUB or GOSUB'

RETURN CLEAR

Example: 100 RETURN CLEAR

Clear internal subroutine tables without executing a RETURN.

ROTATE (alpha variable, d)

Example: 100 ROTATE (A$, 4)

Rotate the bits of each character in the alpha variable d-bits to the left.

$$\textbf{SELECT}\left\{\begin{array}{l}\text{CI xyy}\\ \text{CO xyy [(line length)]}\\ \text{DISK xyy}\\ \text{TAPE xyy}\\ \text{\#f xyy}\\ \text{LIST xyy [(line length)]}\\ \text{PRINT xyy [(line length)]}\\ \text{INPUT xyy}\\ \text{PLOT xyy}\\ \text{P [time]}\\ \text{D or R or G}\end{array}\right\}\text{[, parameter]}\;.$$

Example: 10 SELECT CO 215 (130)

Select device address, line length, file no., pause, degrees, radians or grads.

STOP ["character string"]

Example: 100 STOP "TOUCH CONTINUE TO PROCEED"

Halt program execution and display the STOP line.

TRACE [OFF]

Example: 300 TRACE

Turn trace mode on [or off]

$$\textbf{UNPACK}\text{ (image) }\left\{\begin{array}{l}\text{alpha variable}\\ \text{alpha array desig.}\end{array}\right\}\text{ TO }\left\{\begin{array}{l}\text{numeric variable}\\ \text{numeric array desig.}\end{array}\left[,\left\{\begin{array}{l}\text{numeric variable}\\ \text{numeric array desig.}\end{array}\right\}\right]\ldots\right\}$$

Example: 100 UNPACK (##.#) A$ TO X, A( ), M

Unpack data (PACKed with same image) from alpha variable and store it in numeric variable(s).

$$\textbf{XOR}\left(\text{alpha variable, }\left\{\begin{array}{l}\text{hh}\\ \text{alpha variable}\end{array}\right\}\right)$$

Example: 100 XOR (A$, 2F)

Exclusive OR the second argument into the first.

BASIC FUNCTIONS

FNa (expression)

Example: 200 PRINT X + FNF (2*B)

Reference a function defined in a DEFFN statement with the same identifier (a).

HEX(hh[hh] . . .)

Example: 100 A$ = HEX (4243)

Provide any eight-bit codes in a program.

LEN(alpha variable)

Example: 100 X = LEN (A$)

Determine the number of characters in an alpha variable.

NUM(alpha variable)

Example: 100 N = NUM (K$)

Count the number of ASCII characters in the specified alpha variable which represent a valid number.

$$\textbf{POS}\left(\text{alpha variable }\left\{\begin{array}{l}<\\ <=\\ =\\ >=\\ >\\ <>\end{array}\right\}\left\{\begin{array}{l}\text{"character"}\\ \text{hh}\end{array}\right\}\right)$$

Example: 100 P = POS (A$ = "$")

Find the position of the first character in the alpha variable that satisfies the specified relation.

STR(alpha variable, s [, n])

Example: 100 STR (A$, 4) = STR (B$, 5, 3)

Specify a substring of an alpha variable.

TAB(expression)

Example: 100 PRINT TAB (K + 1); K

Output values in tabulated format.

$$\textbf{VAL}\left(\begin{array}{l}\text{alpha variable}\\ \text{"character string"}\end{array}\right)$$

Example: 100 X = VAL (A$)

Convert the ASCII value of the first character of the argument to an integer.

MATH FUNCTIONS

SIN(expr)

Example: 100 A = SIN(B)

Find the sine of the expression.

COS(expr)

Example: 100 A = COS (B+2)

Find the cosine of the expression.

TAN(expr)

Example: 100 A = TAN (B+C)

Find the tangent of the expression.

ARCSIN(expr)

Example: 100 A = ARCSIN (B)

Find the arcsine of the expression.

ARCCOS(expr)

Example: 100 A = ARCCOS (B)

Find the arccosine of the expression.

ARCTAN(expr) or **ATN**(expr)

Example: 100 A = ARCTAN (B*C)

Find the arctangent of the expression.

EXP(expr)

Example: 100 A = EXP (.33* (5–6))

Find the value of $e^{(expression)}$

INT(expr)

Example: 100 A = INT (3.6)

Find the greatest integer ⩽ value of the expression.

#PI

Example: 100 A = #PI*4

Obtain the constant π.

ABS(expr)

Example: 100 A = ABS (–13.7)

Find the absolute value of the expression.

LOG(expr)

Example: 100 A = LOG (305)

Find the natural logarithm of the expression.

RND(expression)

Example: 100 PRINT RND (I)

Generate a random number $0 < \text{number} < 1$.

SGN(expression)

Example: 100 A = SGN (–.124)

Determine the sign of the expression
(if < 0 = –1; if 0, = 0; if > 0, = 1).

SQR(expr)

Example: 100 A = SQR (18+6)

Find the square root of the expression.

NON-PROGRAMMABLE COMMANDS

```
      ⎡P   [line no. [, line no.] ]⎤
CLEAR ⎢V                           ⎥
      ⎣N                           ⎦
```

Example: **CLEAR P 10, 500**

Clear all or part (P) of program text; all variables (V) or non-common variables (N).

LIST [S] [line no. [, line no.]]

Example: **LIST S 100, 999**

List all or part of program currently in memory.

RENUMBER [line no. 1] [, line no. 2] [, increment]

Example: **RENUMBER 100, 10, 3**

Renumber program lines; [first line to renumber (no. 1), its new line number (no. 2) and increment].

RUN [line no.]

Example: **RUN 35**

Initiate program execution [from specified line]

SORT STATEMENTS

All arguments are array designators; sub-arguments (s, n) refer to bytes; r, c = row, column. Arrays *must* be pre-defined in DIM or COM statements; a = a numeric array, a$ = an alpha array, w$ = work vector, ℓ$ = locator (subscript) array.

MAT CONVERT a( ) TO a$( ) [(s, n)]

Example: **100 MAT CONVERT A( ) TO A$( ) (6, 8)**

Convert elements of numeric array to elements of alpha array in sort format.

MAT COPY [–] a$( ) [<s, n>] TO [–] b$( ) [<s, n>]

Example: **100 MAT COPY A$( ) <X*Y, 100/X> TO –A$( ) < 10, 20>**

Transfer data from first array to second. Element boundaries are ignored.

MAT MERGE a$( ) TO w1$( ), w2$( ), ℓ$( )

Example: **100 MAT MERGE A$ ( ) TO W1$( ), W2$ ( ), S$( )**

Merge pre-sorted rows of a$( ); build a list of pointers in the locator array.

MAT MOVE a$( ) [(s, n)] , ℓ$(r [, c]) [, m] TO b$(r [, c])

MAT MOVE a( ), ℓ$(r [, c]) [, m] TO b (r [, c])

Examples: **100 MAT MOVE A$( ), L$(2, 3), M TO B$(1, 1)**
100 MAT MOVE A( ), L$(1) TO A1(1)

Move data element-by-element from the first array (a( ), a$( )) to the second (b( ), b$( )) as specified by the subscripts of the locator array ℓ$( ).

```
                          ⎧<  ⎫
                          ⎪>  ⎪
MAT SEARCH a$( ) [<s, n>], ⎨<= ⎬  alpha variable   TO b$( ) [STEP expression]
                          ⎪>= ⎪
                          ⎪=  ⎪
                          ⎩<> ⎭
                          └──────────┬──────────┘
                                  relation
```

Example: **100 MAT SEARCH A$( ) < 1, 5>, = STR(Z$, 1, 5) TO B$( ) STEP 5**

Search the first array a$( ) for substrings which satisfy the specified relation; store their positions in the second array b$.

MAT SORT a$( ) TO w$( ), ℓ$( )

Example: **100 MAT SORT A$( ) TO W$( ), L$( )**

Create a locator array (ℓ$( )) of subscripts (pointers) arranged according to the ascending order of the elements in the first alpha array a$( ).

MATRIX STATEMENTS

a, b, c are numeric array names
a$, b$ are alpha array names

MAT a = a + b

Example: **100 MAT A = A + D**

Add two matrices.

MAT a = CON [(expr 1 [, expr 2])]

Example: **100 MAT A = CON(10)**

Set all elements of the specified array equal to one; [redimension the array with the expressions].

MAT a = b

Example: **100 MAT A = B**

Replace each element of array a with the corresponding element of array b.

MAT a = IDN [(expr 1 [, expr 2])]

Example: **100 MAT B = IDN (5, 5)**

Set the specified (square) array equal to the identity matrix; [redimension it with the expressions].

```
          ⎧a[(expr 1 [, expr 2] )]                   ⎫⎡,  ⎧b [(expr 1 [, expr 2] )]                  ⎫⎤
MAT INPUT ⎩a$ [(expr 1 [, expr 2] ) [length] ]       ⎭⎣,  ⎩b$ [(expr 1 [, expr 2] ) [length] ]       ⎭⎦  . .
```

Example: **100 MAT INPUT A, B(2), C$ (2, 2) 4**

Accept values for array elements from keyboard (or other input-class device); [redimension].

MAT a = INV (b) [, numeric scalar]

Example: **100 MAT B = INV(A), D**

Invert a matrix [obtain determinant].

MAT a = b*c

Example: **100 MAT A = B*C**

Store the product of two arrays in the first array (a).

MAT PRINT array name [t array name] [t]

Example: **100 MAT PRINT A, B$; C$**

Print or display array elements row by row, in zoned (t=,) or packed (t=;) format.

```
         ⎧a [(expr 1 [, expr 2] )]                ⎫⎡, ⎧b [(expr 1 [, expr 2] )]                ⎫⎤
MAT READ ⎩a$ [(expr 1 [, expr 2] ) [length] ]     ⎭⎣, ⎩b$ [(expr 1 [, expr 2] ) [length] ]     ⎭⎦  . . .
```

Example: **100 MAT READ A, B(2, 3), D$(4) 6**

Assign values from DATA statement(s) to all array elements; [redimension].

```
          ⎧a (expr 1 [, expr 2]                   ⎫  ⎡,   ⎧b (expr 1 [, expr 2] )                  ⎫⎤
MAT REDIM ⎩a$ (expr 1 [, expr 2] ) [length]       ⎭  ⎣,   ⎩b$ (expr 1 [, expr 2] ) [length]        ⎭⎦  . . .
```

Example: **100 MAT REDIM A(4,5), B$(2,3)4**

Redimension the specified arrays (to not more than the no. of bytes in the original array).

MAT a = (expr)*b

Example: **100 MAT A = (5)*A**

Perform scalar multiplication on array b; assign product to array a.

MAT c = a – b

Example: **100 MAT C = A – B**

Subtract matrices of the same dimension.

MAT c = TRN (a)

Example: **100 MAT C = TRN (A)**

Replace array c with the transpose of array a.

MAT c = ZER [(expr 1 [, expr 2])]

Example: **100 MAT A = ZER (F, T + 2)**

Set all elements of specified array to zero; [redimension]

GENERAL I/O STATEMENTS

$GIO [comment] $\begin{bmatrix}\#f\\ /xyy\end{bmatrix}$ (microcommand sequence, error/status register) [data buffer]

Examples: 100 $GIO WRITE/03B (6C01 4400 A206 8601, R$) B$( )

10 DIM M$8
20 M$ = HEX(6C0744008601)
100 $GIO (M$, R$)

Perform the I/O operation specified by the microcommand sequence; flag errors in the error/status register; use data buffer for input/output data transfer.

$IF ON $\begin{bmatrix}\#f,\\ /xyy,\end{bmatrix}$ line no.

Example: 100 $IF ON #3, 200

Test ready condition of I/O device (data-ready for input unit, device-ready for output unit); branch to specified line number if condition = ready.

$TRAN (data block, table or list) $\underbrace{[hh]}_{\text{mask}}$ [R]

Example: 100 $TRAN (X$, T$( )) 2F

Translate all characters in data block according to table or list [mask bytes (AND with hh) or replace with list characters (R)].

$PACK $\underbrace{\left[\left\{\begin{matrix}D = a\\ F = a\end{matrix}\right\}\right]}_{\text{format specifier}}$ data buffer FROM variable [, variable] . . .

Examples: 90 D1$ = HEX(002C)
100 $PACK (D = D1$) B1$( ) FROM X( ), Y$

Transfer data from the variable(s) and pack it in the data buffer according to the format specified.

$UNPACK $\underbrace{\left[\left(\begin{matrix}D = a\\ F = a\end{matrix}\right)\right]}_{\text{format specifier}}$ data buffer TO variable [, variable] . .

Example: 100 $UNPACK (F = F2$) B2$( ) TO M, P$, Q( )

Separate data stored in the data buffer according to the format specified, convert and transfer it to the variables given.

PLOTTER STATEMENTS

PLOT [expr 0] < [expr 1], [expr 2], $\underbrace{\begin{bmatrix}\text{U}\\ \text{D}\\ \text{S}\\ \text{C}\\ \text{R}\\ \text{“literal string”}\\ \text{avar}\end{bmatrix}^{*}}_{\text{plot arg}}$ > [, < [expr 1], [expr 2], [plot arg] >] . .

Example: 100 PLOT < X, Y, "VALUE" >, < 40, 60, A$ >

Move plotting carrier Δx (expr 1) and Δy (expr 2) increments; plot [expr 0] times according to plot argument.

*U = pen up; D = pen down; S = set horizontal (expr 1) and vertical (expr 2) spaces; C = set character size; R = reset to zero; literal string, avar = plot alphanumeric labels.

ABBREVIATIONS

al array	=	alpha array
arg	=	argument
a var	=	alpha variable
desig	=	array designator (e.g., A( ), A$( ))
elt	=	array element
expr	=	numeric expression
f	=	file number assigned with SELECT statement
hh	=	hexcode
int	=	integer
n array	=	numeric array
no.	=	number
n var	=	numeric variable
var	=	variable (numeric or alpha scalar, numeric or alpha array, STR function (a var), literal in ('')
xyy	=	device address

Printed in U.S.A. 700-3500F 7-80-2M

TAPE CASSETTE UNITS STATEMENTS

BACKSPACE $\begin{bmatrix}\text{\# f,}\\ \text{/xyy,}\end{bmatrix}\ \begin{Bmatrix}\text{BEG}\\ \text{n}\\ \text{nF}\end{Bmatrix}$

Example: **100 BACKSPACE 5 F**

Backspace to beginning of file (BEG), n logical records or n files (nF).

DATALOAD $\begin{bmatrix}\text{\#f,}\\ \text{/xyy,}\end{bmatrix}\ \begin{Bmatrix}\text{``name''}\\ \text{variable [, variable] . .}\end{Bmatrix}$

Example: **100 DATALOAD A, B( ), C$**

Locate a data file by name, or load a logical data record.

DATALOAD BT $\begin{bmatrix}\text{(N = expr)}\end{bmatrix}\ \begin{bmatrix}\text{\#f,}\\ \text{/xyy,}\end{bmatrix}$ alpha array desig.

Example: **100 DATALOAD BT (N = 100) #5, Q$( )**

Load a block of N (100 or 256) bytes of data.

DATARESAVE $\begin{bmatrix}\text{\#f,}\\ \text{/xyy,}\end{bmatrix}\ \begin{Bmatrix}\text{OPEN ``name''}\\ \text{variable [, variable] . .}\end{Bmatrix}$

Example: **100 DATARESAVE A, B$**

Rewrite a header or a data record, in place.

DATASAVE $\begin{bmatrix}\text{\#f,}\\ \text{/xyy,}\end{bmatrix}\ \begin{Bmatrix}\text{OPEN ``name''}\\ \text{END}\\ \text{variable [, variable] . . .}\end{Bmatrix}$

Example: **100 DATASAVE OPEN "FOXI"**

Write a data header (OPEN . . .), trailer (END) or record (variable . . .).

DATASAVE BT (R] [(([N = expr [,]] (H])] $\begin{bmatrix}\text{\#f,}\\ \text{/xyy,}\end{bmatrix}$ alpha array desig.

Example: **100 DATASAVE BT R (N = 100, H)/10C, A$( )**

Record [or resave (R)] an N-byte (100 or 256) data block [or a header record (H)] .

IF END THEN line no.

Example: **100 IF END THEN 130**

If End of File record is encountered, branch to the specified line number.

LOAD $\begin{bmatrix}\text{\#f}\\ \text{/xyy}\end{bmatrix}$ [[,] "name"] [line no. 1] [, line no. 2]

Example: **100 LOAD "SAM" 400**

Load a program file [by name].

REWIND $\begin{bmatrix}\text{\#f}\\ \text{/xyy}\end{bmatrix}$

Example: **100 REWIND/10C**

Rewind cassette [in indicated unit] .

SKIP $\begin{bmatrix}\text{\#f,}\\ \text{/xyy,}\end{bmatrix}\ \begin{Bmatrix}\text{END}\\ \text{n}\\ \text{nF}\end{Bmatrix}$

Example: **100 SKIP 2F**

Skip n logical records, n files (nF), or to trailer record on data file (END).

COMMANDS

LOAD $\begin{bmatrix}\text{\#f}\\ \text{/xyy}\end{bmatrix}$ [[,] "name"]

Example: **LOAD/10B, "PROG A"**

Load a program file [by name].

SAVE $\begin{bmatrix}\text{\#f,}\\ \text{/xyy,}\end{bmatrix}$ [P] ["name"] [line no. [, line no.]]

Example: **SAVE #2, "LINDY"**

Record all [or part] of a program [by name] ; [protect program (P)] .

DISK UNITS STATEMENTS

#f in disk statements can be an integer (1 to 6) or a numeric variable; sector addresses can be expressions or alpha variables.

COPY $\begin{bmatrix}\text{\#f,}\\ \text{/xyy,}\end{bmatrix}\ \begin{Bmatrix}\text{FR}\\ \text{RF}\end{Bmatrix}$ (expr 1, expr 2)

Example: **COPY FR(0, 1023)**

Copy the sectors specified from one platter to another.

DATALOAD BA $\begin{Bmatrix}\text{F}\\ \text{R}\\ \text{T}\end{Bmatrix}\ \begin{bmatrix}\text{\#f,}\\ \text{/xyy,}\end{bmatrix}\ \left(\text{sector address, }\begin{Bmatrix}\text{L}\\ \text{LS}\end{Bmatrix}\right)$ alpha array desig.

Example: **100 DATALOAD BA F (20,L) A$( )**

Read one sector (including control information) into the specified array.

DATALOAD DA $\begin{Bmatrix}\text{F}\\ \text{R}\\ \text{T}\end{Bmatrix}\ \begin{bmatrix}\text{\#f,}\\ \text{/xyy,}\end{bmatrix}\ \left(\text{sector address, }\begin{Bmatrix}\text{L}\\ \text{L\$}\end{Bmatrix}\right)$ variable [, variable] .

Example: **100 DATALOAD DA F/320, (D,D) A, B**

Read a logical record starting at the sector address and assign the values to the specified variables.

DATALOAD DC [#f,] variable [, variable] .

Example: **100 DATALOAD DC S( ), A$**

Read the next logical record from a cataloged disk file and assign the values to the variables.

DATALOAD DC OPEN $\begin{Bmatrix}\text{F}\\ \text{R}\\ \text{T}\end{Bmatrix}$ [#f,] $\begin{Bmatrix}\text{TEMP, expr 1, expr 2}\\ \text{``name''}\end{Bmatrix}$

Example: **100 DATALOAD DC OPEN F #A, "LIST"**

Open a previously cataloged data file (set up file sector addresses).

DATASAVE BA $\begin{Bmatrix}\text{F}\\ \text{R}\\ \text{T}\end{Bmatrix}$ [$] $\begin{bmatrix}\text{\#f,}\\ \text{/xyy,}\end{bmatrix}\ \left(\text{sector address, }\begin{Bmatrix}\text{L}\\ \text{L\$}\end{Bmatrix}\right)$ alpha array desig.

Example: **100 DATASAVE BAT#2, (Q,Q) D$( )**

Write one sector from the array with no control information.

DATASAVE DA $\begin{Bmatrix}\text{F}\\ \text{R}\\ \text{T}\end{Bmatrix}$ [$) $\begin{bmatrix}\text{\#f,}\\ \text{/xyy,}\end{bmatrix}\ \left(\text{sector address, }\begin{Bmatrix}\text{L}\\ \text{L\$}\end{Bmatrix}\right)\ \begin{Bmatrix}\text{END}\\ \text{variable [, variable] . .}\end{Bmatrix}$

Example: **100 DATASAVE DA F(2∗K, L) J( ), K1**

Write one logical record (one or more sectors) from variables starting at the sector address, or write a trailer record (END).

DATASAVE DC [$] [#f,] $\begin{Bmatrix}\text{END}\\ \text{variable [, variable] . .}\end{Bmatrix}$

Example: **100 DATASAVE DC $#4, END**

Write one record containing all variables specified starting at the current file position, or write a trailer record (END).

DATASAVE DC CLOSE $\begin{bmatrix}\text{\#f}\\ \text{ALL}\end{bmatrix}$

Example: **100 DATASAVE DC CLOSE ALL**

Close (clear sector address data of) the specified file or all files.

DATASAVE DC OPEN $\begin{Bmatrix}\text{F}\\ \text{R}\\ \text{T}\end{Bmatrix}$ [$] [#f,] $\left\{\begin{matrix}\begin{Bmatrix}\text{name 1}\\ \text{expr}\end{Bmatrix}\text{ , ``name 2''}\\ \text{TEMP, expr 1, expr 2}\end{matrix}\right\}$

Example: **100 DATASAVE DC OPEN F$ 200, "FIL1"**

Reserve space in the Catalog for a cataloged data file, or for a temporary work file outside the catalog area.

DBACKSPACE $\text{[\#f,]}\ \left\{\begin{matrix}\text{BEG}\\\text{expr (S)}\end{matrix}\right\}$

Example: 100 DBACKSPACE #2, 5S

Backspace number of records or sectors (S), or to starting sector address of file (BEG).

DSKIP $\text{[\#f,]}\ \left\{\begin{matrix}\text{END}\\\text{expr [S]}\end{matrix}\right\}$

Example: 100 DSKIP 4+K

Skip the number of logical records [or sectors (S)] specified, or to the end of file (END).

IF END THEN line no.

Example: 100 IF END THEN 700

Branch to line number if an end of file (trailer) was encountered on the last disk operation.

LIMITS $\left\{\begin{matrix}\text{F}\\\text{R}\\\text{T}\end{matrix}\right\}\ \text{[\#f,] [name,] variable 1, variable 2, variable 3}$

Example: 100 LIMITS F "NONAME", A, B, C

Obtain starting address, ending address and number of sectors (or current sector address) of file.

LIST DC $\left\{\begin{matrix}\text{F}\\\text{R}\\\text{T}\end{matrix}\right\}\ \left[\begin{matrix}\text{\#f}\\\text{/xyy}\end{matrix}\right]$

Example: LIST DC F

List the contents of the catalog index.

LOAD DA $\left\{\begin{matrix}\text{F}\\\text{R}\\\text{T}\end{matrix}\right\}\ \left[\begin{matrix}\text{\#f,}\\\text{/xyy,}\end{matrix}\right]\ \left(\text{sector address, }\underbrace{\left\{\begin{matrix}\text{L}\\\text{L\$}\end{matrix}\right\}}_{\text{a common variable}}\right)\ \text{[line no. 1 [, line no. 2]]}$

Example: 100 LOAD DA F (40, L) 400

Load and execute program from the specified starting sector address.

LOAD DC $\left\{\begin{matrix}\text{F}\\\text{R}\\\text{T}\end{matrix}\right\}\ \left[\begin{matrix}\text{\#f,}\\\text{/xyy,}\end{matrix}\right]\ \text{name [line no. 1 [, line no. 2]]}$

Example: 100 LOAD DC R "PROG1" 100, 200

Load the specified cataloged program [or segment] into memory and execute it.

MOVE $\left[\begin{matrix}\text{\#f,}\\\text{/xyy,}\end{matrix}\right]\ \left\{\begin{matrix}\text{FR}\\\text{RF}\end{matrix}\right\}$

Example: 100 MOVE FR

Compress catalog and move it to another platter.

MOVE END $\left\{\begin{matrix}\text{F}\\\text{R}\\\text{T}\end{matrix}\right\}\ \left[\begin{matrix}\text{\#f}\\\text{/xyy}\end{matrix}\right]\ \text{= expr}$

Example: MOVE END F = 4799

Move the end of the catalog area to the sector specified by expression.

SCRATCH $\left\{\begin{matrix}\text{F}\\\text{R}\\\text{T}\end{matrix}\right\}\ \left[\begin{matrix}\text{\#f,}\\\text{/xyy,}\end{matrix}\right]\ \text{name [, name] . . .}$

Example: 100 SCRATCH F "WORKFILE"

Change the file status of the named disk files to 'scratched'.

SCRATCH DISK $\left\{\begin{matrix}\text{F}\\\text{R}\\\text{T}\end{matrix}\right\}\ \left[\begin{matrix}\text{\#f,}\\\text{/xyy,}\end{matrix}\right]\ \text{[LS = expr 1,] END = expr 2}$

Example: 100 SCRATCH DISKF#2, LS = 4, END = 1023

Reserve space on disk [for the Catalog Index (expr 1)] and the Catalog Area (expr 2 = last sector to be used).

VERIFY $\left\{\begin{matrix}\text{F}\\\text{R}\\\text{T}\end{matrix}\right\}\ \left[\begin{matrix}\text{\#f,}\\\text{/xyy,}\end{matrix}\right]\ \text{(expr 1, expr 2)}$

Example: VERIFY R (0, 1023)

Verify data in specified range of sectors.

DISK UNITS COMMANDS

LOAD DA $\left\{\begin{matrix}\text{F}\\\text{R}\\\text{T}\end{matrix}\right\}\ \left[\begin{matrix}\text{\#f,}\\\text{/xyy,}\end{matrix}\right]\ \left(\text{sector address, }\left\{\begin{matrix}\text{L}\\\text{LS}\end{matrix}\right\}\right)$

Example: LOAD DAF (24, D)

Load the program from the specified starting sector.

LOAD DC $\left\{\begin{matrix}\text{F}\\\text{R}\\\text{T}\end{matrix}\right\}\ \left[\begin{matrix}\text{\#f,}\\\text{/xyy,}\end{matrix}\right]\ \text{name}$

Examples: LOAD DCFA$
LOAD DCF "LIST"

Load the specified cataloged program file.

SAVE DA $\left\{\begin{matrix}\text{F}\\\text{R}\\\text{T}\end{matrix}\right\}\ \text{[S]}\ \left[\begin{matrix}\text{\#f,}\\\text{/xyy,}\end{matrix}\right]\ \text{[P]}\ \left(\text{sector address, }\left\{\begin{matrix}\text{L}\\\text{LS}\end{matrix}\right\}\right)\text{[line no. 1 [, line no. 2]]}$

Example: SAVE DAF$/320, P(L, L) 100, 200

Save all or part of a program on disk, starting at the specified sector.

SAVE DC $\left\{\begin{matrix}\text{F}\\\text{R}\\\text{T}\end{matrix}\right\}\ \text{[\$]}\ \left[\left(\begin{matrix}\text{expr}\\\text{name 1}\end{matrix}\right)\right]\ \left[\begin{matrix}\text{\#f,}\\\text{/xyy,}\end{matrix}\right]\ \text{[P] name 2 [line no. 1 [, line no. 2]]}$

Example: SAVE DCR (100) #2, "CONVERT"/SAVE DCF ("old") "new"

Save all or part of a program on disk and catalog it.

PUNCHED TAPE READER AND TELETYPE® STATEMENTS

DATALOAD $\left[\begin{matrix}\text{\#f,}\\\text{/xyy,}\end{matrix}\right]\ \text{variable [, variable] .}$

Examples: (reader): 100 DATALOAD/618, A1$( ), A2
(Teletype): 100 DATALOAD/41D, A1$( ), X, STR(A$, 2)

Read values from punched tape and assign them to variables given.

DATALOAD BT $\underset{\substack{\uparrow\\\text{(not valid}\\\text{on Teletype)}}}{\text{[R]}}\left[\left(\text{[N = expr] [,] [L = }\left\{\begin{matrix}\text{hh}\\\text{alpha var}\end{matrix}\right\}\text{][,] [S = }\left\{\begin{matrix}\text{hh}\\\text{alpha var}\end{matrix}\right\}\text{]}\right)\right]\left[\begin{matrix}\text{\#f,}\\\text{/xyy,}\end{matrix}\right]\ \underbrace{\left\{\begin{matrix}\text{al. var.}\\\text{al. arr. desig.}\end{matrix}\right\}}_{\text{variables}}$

Examples: (reader): 100 DATALOAD BT (N = 200, L = 00, S = A$)/618, B$( )
Teletype): 100 DATALOAD BT (N = 3, L = FF, S = 0D)/41D, A$

Read punched tape [backwards (R)] and store values read in the specified variable or array. [Read N characters, ignore Leader code characters (L), stop when stop character (S) is encountered.]

LOAD $\left[\begin{matrix}\text{\#f}\\\text{/xyy}\end{matrix}\right]\ \text{[[,] line no. 1] [, line no. 2]}$

Examples: (reader): 100 LOAD #2
(Teletype): 100 LOAD/41D, 100

Load program [by line number] into memory and execute it (clears previous program).

COMMAND

LOAD $\left[\begin{matrix}\text{\#f}\\\text{/xyy}\end{matrix}\right]$

Examples: (reader): LOAD/618
(Teletype): LOAD/41D

Load program from punched tape, add it to program currently in memory.

CARD READERS STATEMENTS

DATALOAD $\left[\begin{matrix}\text{\#f,}\\\text{/xyy,}\end{matrix}\right]\ \text{variable [, variable] .}$

Example: 100 DATALOAD #3, N( ), A$, N

Read card(s) and assign the values to variables.

DATALOAD BT $\left[\left(\text{[N = expr] [,]}\ \left[\text{L = }\begin{Bmatrix}\text{hh}\\ \text{al var}\end{Bmatrix}\right]\ \text{[,]}\ \left[\text{S = }\begin{Bmatrix}\text{hh}\\ \text{al var}\end{Bmatrix}\right]\right)\right]\begin{bmatrix}\text{\#f,}\\ \text{/xyy,}\end{bmatrix}\begin{Bmatrix}\text{al var}\\ \text{al array desig.}\end{Bmatrix}$

(mark sense
card reader)

Example: **100 DATALOAD BT (N = 40, L = FF, S = 99)/517, B$**

Read [up to N] characters [ignore leader characters (L=), stop at stop character (S=)], store them in the alpha scalar or array.

DATALOAD BT (N = 82) $\begin{bmatrix}\text{\# f,}\\ \text{/xyy,}\end{bmatrix}$ alpha array desig.

(punched card
reader)

Example: 100 DATALOAD BT (N = 82)/629, B$( )

Read 80-character Hollerith card images, translate them to ASCII codes and store them in the array.

DATASAVE BT $\begin{bmatrix}\text{\#f,}\\ \text{/42E,}\end{bmatrix}$ alpha variable

(punched
card reader)

Example: **100 DATASAVE BT/42E, F$**

Initiate reading of subsequent card in hopper into card reader buffer (used before DATALOAD, DATALOAD BT statements).

INPUT ["character string",] variable [, variable] .

Example: **100 INPUT "ENTER A AND B", A, B**

Read data values (separated by commas) from card(s) and assign them to variables in the INPUT statement.

LOAD $\begin{bmatrix}\text{\#f}\\ \text{/62B}\end{bmatrix}$ [[,] line no. 1, line no. 2]

(punched
card reader)

Example: **100 LOAD/62B, 100, 200**

Load a BASIC program from punched cards into memory with conversion from Hollerith to ASCII code.

COMMANDS

SELECT $\begin{Bmatrix}\textbf{CI}\\ \textbf{INPUT}\end{Bmatrix}$ xyy

(mark sense
card reader)

Example: SELECT CI 517

Select the card reader as the Console Input (CI) device to allow card reader to run in batch mode. (When loading is complete, reselect keyboard as CI device with SELECT CI 001 on last card.)

LOAD $\begin{bmatrix}\text{\#f}\\ \text{/62B}\end{bmatrix}$

(punched
card reader)

Example: LOAD/62B

Load Hollerith program card text, convert it to ASCII and add it to the program in memory.

ERROR CODES

CODE	MEANING
01	Text Overflow (insufficient memory for program text)
02	Table Overflow (insufficient memory for internal tables, subroutine and FOR/NEXT loops; non-common variables are cleared)
03	Math Error
04	Missing Left Parenthesis
05	Missing Right Parenthesis
06	Missing Equal Sign
07	Missing Quotation Marks (")
08	Undefined FN Function Encountered
09	More than Five levels of FN Nesting Encountered
10	Incomplete Statement
11	Missing Line Number or Illegal Attempt to CONTINUE
12	Missing Statement Text
13	Missing or Illegal Integer
14	Missing Relational Operator
15	Missing Expression
16	Missing Scalar Variable
17	Missing Array Variable
18	Illegal Value
19	Missing Numeric Value
20	Illegal Format of Number
21	Missing Letter or Digit
22	Array Variable Not Defined
23	No Program Statements Found
24	Illegal Immediate Mode Statement
25	Illegal GOSUB/RETURN Use
26	Illegal FOR/NEXT Use
27	Insufficient Data for READ Statement
28	RESTORE Reference Out of Bounds
29	Illegal Data Format for INPUT Statement
30	Illegal Placement of COM Statement
31	Line Number > 9999 Generated
33	Missing Hexdigit
34	Tape Cassette Read Error
35	Missing Comma or Semicolon
36	Format (##...) Missing From Image (%) Statement
37	Statement Referenced is not an Image Statement
38	Exponential Format Missing Four ↑'s
39	Missing Literal in Quotes ("")
40	Missing Alpha Variable
41	Illegal Argument in STR(Function
42	Program Name > 8 Characters
43	Wrong Variable Type
44	Program Protected (cannot be SAVEd or LISTed)
45	Program Line > 192 Keystrokes
46	Incorrect RENUMBER Use
47	#f File Specification not SELECTed
48	Special Function Key Not Defined with DEFFN'
49	End of Cassette Tape
50	Tape Cassette Tab Removed
51	Illegal Statement
52	Expected Data Record
53	Illegal Use of HEX Function
54	Illegal Argument in PLOT Statement
55	Illegal Argument in BT Statement
56	Too Few #'s in Image
57	Illegal Value (Disk Sector Address or MAT Array)
58	Expected Data Record
59	Illegal Alpha Variable
60	Array Too Small
61	Disk Timing Error
62	Disk File Full
63	Missing Alpha Array Designator
64	Sector Not on Disk or Disk Not Scratched
65	Disk Error (Medium or Hardware)
66	Disk Format Key Engaged
67	Disk Format Error
68	Disk LRC Error – Probably Hardware
71	Sector Not Found – Disk Seek Error (Hardware or Medium)
72	Disk CRC Error – Usually Hardware, Possibly Medium
73	Incorrect Syntax Using File Name
74	Catalog End Error
75	Non-Programmable Command
76	PLOT Statement Angle Brackets < > Missing
77	Disk Starting Sector Address > Ending Address
78	Disk File Not Scratched
79	Disk File Already Cataloged
80	Disk File Not in Catalog
81	Illegal /xyy Specification
82	No End-of-File on Disk
83	Disk Hardware Error – Transient or Permanent
84	Not Enough Memory for MOVE or COPY on Disk
85	Disk Read after Write Error
86	Disk File Not Open
87	Common Variable Required for LOAD DA Statement
88	Disk Library Index Full
89	Matrix Not Square
90	Matrix Operands Not Compatible
91	Illegal Matrix Operand
92	Redimensioned Matrix Too Large
93	Matrix is Singular
94	Asterisk Missing in Matrix Statement
95	Illegal Gen I/O Microcommand or Field/Delimiter Specification
96	Missing Gen I/O Buffer
97	Gen I/O Variable or Array Too Small
98	Illegal Gen I/O Arguments to Modify Array
=1	Missing Sort Numeric Array Designator
=2	Sort Array Too Small
=3	Illegal Sort Array Dimensions

SYSTEM ERROR! Unrecoverable. Either machine failure or programmer faux pas
(Run Mode Omitted).