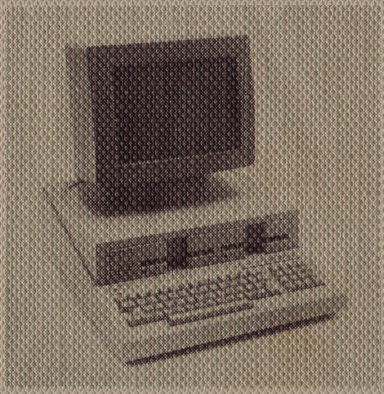
M20

BASIC & PCOS
Pocket Reference



olivetti L1

INTRODUCTION

This guide lists the commands (PCOS and BASIC), edit mode subcommands, BASIC statements, and built-in functions of the Olivetti L1 M20 microcomputer, and provides a general description of each. Within each category, items are presented alphabetically. The guide also includes information on the use of the Control key, a list of error messages with explanations, and the ASCII table shown with numeric equivalences for each of its characters.

The guide is intended for reference only and assumes a familiarity with PCOS, the BASIC language, and the M20. For a complete description of the system, the language, and the M20, see the publications:

L1 M20
Professional Computer
Operating System (PCOS)
User Guide
code 3987590 U (0)

L1 M20
BASIC Language
Reference Guide
code 3982430 P (1)

Distribution: General (G)

1st Edition: August 1982

Release: 1.2

PUBLICATION ISSUED BY:

Ing. C. Olivetti & C., S.p.A.
Servizio Centrale Documentazione
77, Via Jervis - 10015 Ivrea (Italy)

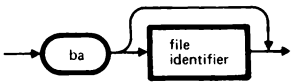
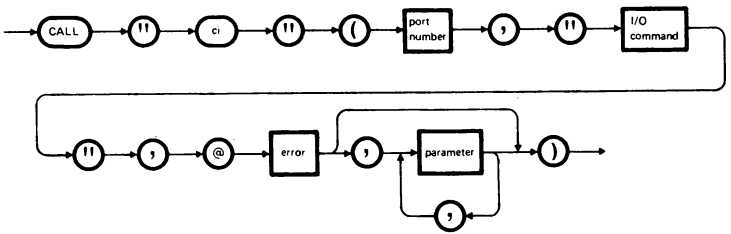
TABLE OF CONTENTS

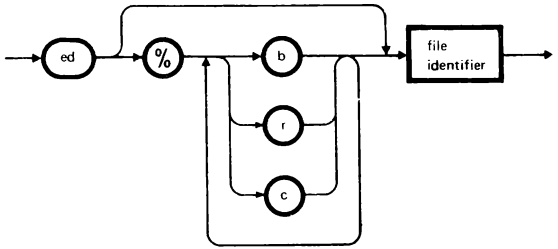
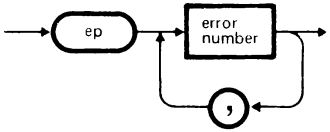
Notation Conventions

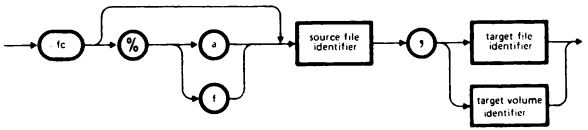
Syntax diagrams are used to describe the commands, statements, and functions in this guide. A syntax diagram is a flowchart with one entry and one exit. Each path through the flowchart shows an allowable sequence of symbols. The diagrams are read from left to right.

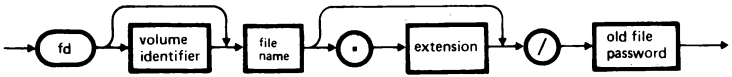
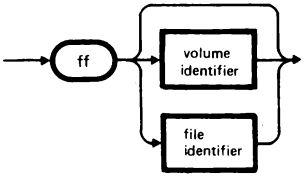
. Items enclosed in ovals or circles, if selected, must be entered exactly as shown.

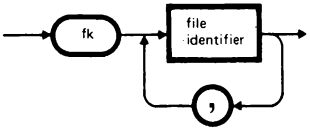
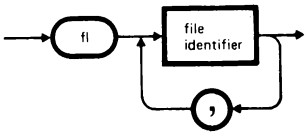
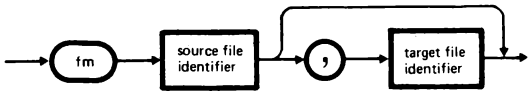
. Items enclosed in rectangles, if selected, must be replaced by information supplied by the user.

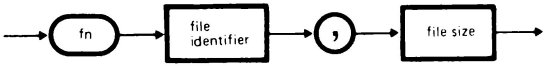
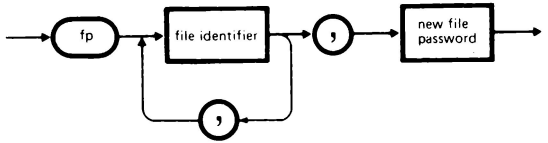
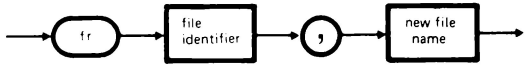
NAME	FUNCTION	FORMAT
BASIC	Invokes the BASIC interpreter and, optionally, executes a program	 Example: ba chem
CI	Allows the use, via a BASIC CALL, of the RS-232-C driver	 Example: 70 CALL "ci" (com:,"r",@error%,@string$)

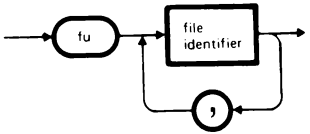
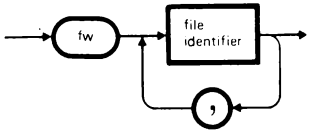
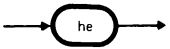
NAME	FUNCTION	FORMAT
EDIT	Invokes the Video File Editor	 Example: ed %c nautilus Note: %b = backup %r = read-only %c = BASIC
EPRINT	Displays explanations for specified error numbers	 Example: ep 58,60

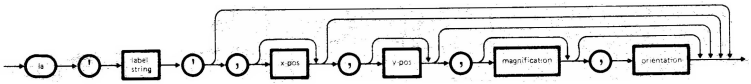
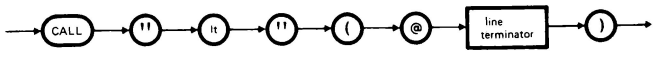
NAME	FUNCTION	FORMAT
FCOPY (Case 1)	Copies the contents of one file into another, optionally appending the first to the second	 Examples: fc 1:exp, 1:test fc 0:ter, 1:kdr Note: The flag "%a" means append; the flag "%f" means force copy (overwrite without confirmation)
FCOPY (Case 2)	Copies one or more files (specified using wild cards) from one disk onto another	 Example: fc 0:*.cmd, 1: Note: The process will skip any copy protected files if the flag "%u" is specified

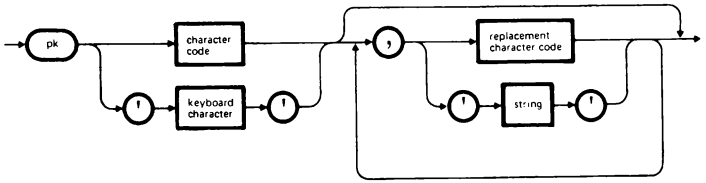
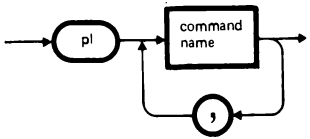
NAME	FUNCTION	FORMAT
FDEPASS	Removes a password previously assigned to a file	 Example: fd max/eagle
FFREE	Frees the unused sector of a file, a group of files, or a volume	 Example: ff lou:

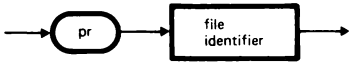
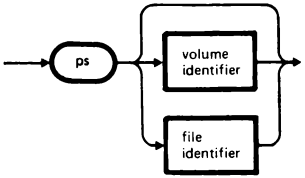
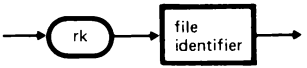
NAME	FUNCTION	FORMAT
FKILL	Deletes one or more files	 Examples: fk minnie:maus fk data:*
FLIST	Displays the contents of one or more files	 Examples: fl chap2/ken fl 0:t*
FMOVE	Copies a file from one disk to another, using only one drive	 Example: fm data2,data2

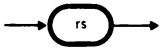
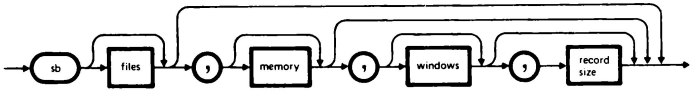
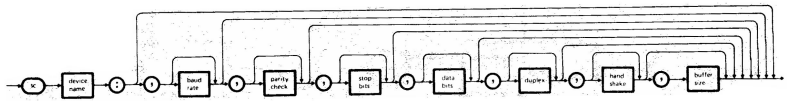
NAME	FUNCTION	FORMAT
FNEW	Allocates space for a file	 Example: fn ver2, 6500
FPASS	Assigns a password to one more files	 Examples: fp tony/waha, alys fp 1:*,emerson
FRENAME	Changes the name of existing file	 Example: fr vers1, final

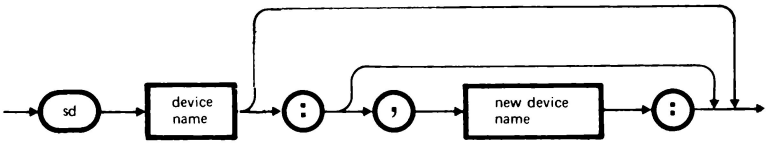
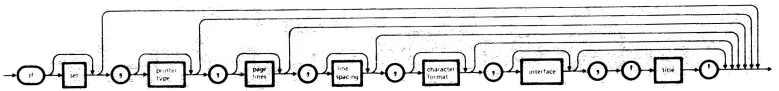
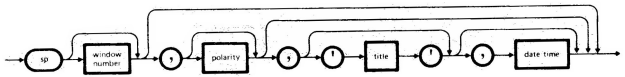
NAME	FUNCTION	FORMAT
FUNPROT	Unprotects one or more write-protected files	 Examples: fu rocar fu 1:*
FWPROT	Write-protects one or more files	 Examples: fw perole fw 1:*
HELP	Invokes HELP display for guidance	 Example: he

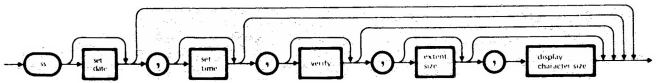
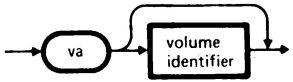
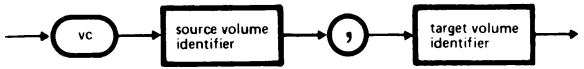
NAME	FUNCTION	FORMAT
IEEE	Loads and initializes the IEEE 488 extension package	 Example: ie
LABEL	Displays a string of characters at a specified location	 Example: la 'year'
LTERM	Returns a value (0,1,2) indicating which of the line terminator keys (↵ , S1 , S2) was last us-d	 Example: 50 CALL "lt" (@1%) Note: LTERM can be used only in a program

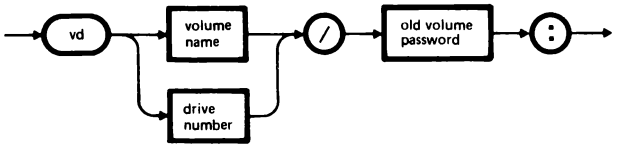
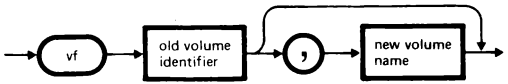
NAME	FUNCTION	FORMAT
PKEY	Assigns a string of characters to a specified key	 Examples: pk &41, &42 pk '#','ba',13,'files',13
PLOAD	Loads a disk-based utility into the "permanent memory area"	 Example: pl vc

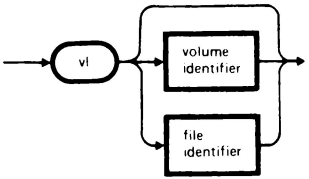
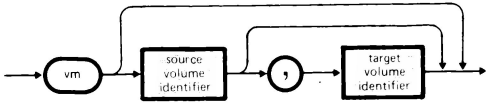
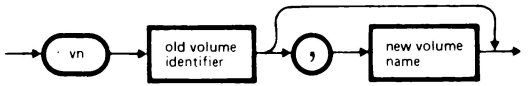
NAME	FUNCTION	FORMAT
PRUN	Calls and initializes an alternative operating system previously saved via a PSAVE command	 Example: pr alt.1
PSAVE	Saves the current PCOS configuration on a disk	 Example: ps
RKILL	Recovers, if possible, the contents of a previously deleted file	 Example: rk ver3/c

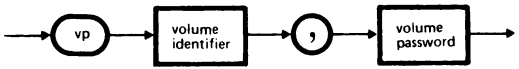
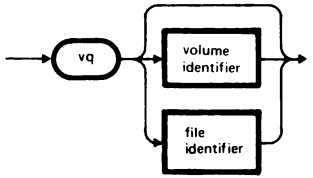
NAME	FUNCTION	FORMAT
RS232	Loads and initializes the RS-232 extension package	 Example: rs
SBASIC	Sets the BASIC environment for programming	 Example: sb 4, 32500, 4, 128
SCOMM	Sets the transmission environment of an RS-232 communications port	 Example: sc com:,,,6

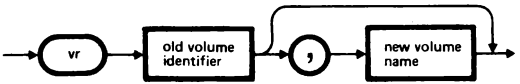
NAME	FUNCTION	FORMAT
SDEVICE	Displays the device table and, optionally, renames a device	 Example: sd cons:,keybd.video: Note: See "Device Rerouting"
SFORM	Specifies the type of printer being used and the printing format	 Example: sf on,,50
SPRINT	Prints the image of either the screen or a specified window	 Example: sp

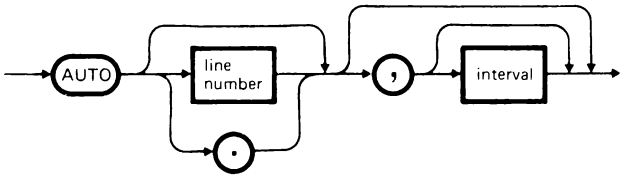
NAME	FUNCTION	FORMAT
SSYS	Sets the global system environment	 Example: ss 06/06/82
VALPHA	Sorts a volume directory in alphabetical order	 Example: va 1:
VCOPY	Copies the contents of one disk onto another	 Example: vc 0:, 1:

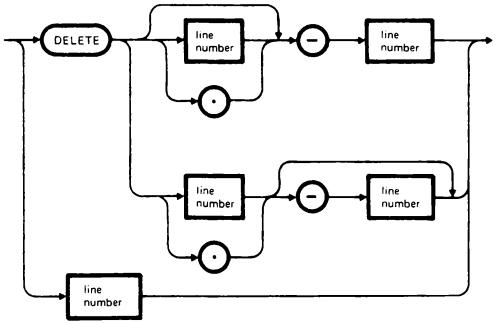
NAME	FUNCTION	FORMAT
VDEPASS	Deletes a disk password	 Example: vd ksqrs/tonyken:
VFORMAT	Performs a hardware format of a disk, creates a blank file system, and allows the user to assign a volume name	 Example: vf rd:, payroll/newyork:

NAME	FUNCTION	FORMAT
VLIST	Lists part or all of a directory	 Example: vl codes/kasparov:
VMOVE	Copies a volume from disk to disk using only one drive	 Example: vm mydisk:, newdisk/pass:
VNEW	Creates a blank file system and allows the user to assign a volume name	 Example: vn exper

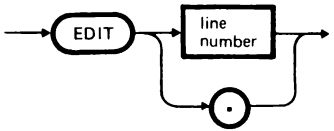
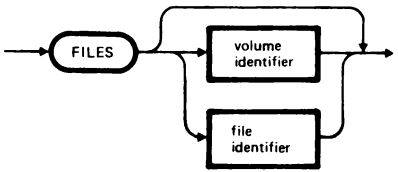
NAME	FUNCTION	FORMAT
VPASS	Assigns a password to a disk or changes an existing one	 Examples: vp eco:,stat vp pay/wknd:,ovrt
VQUICK	Lists the names of files resident on a disk	 Example: vq 1:

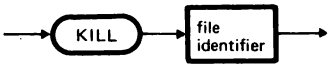
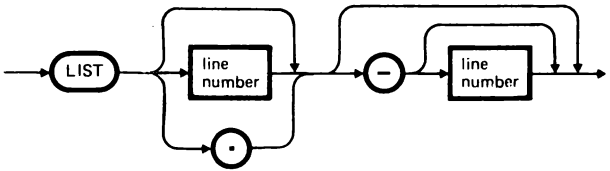
NAME	FUNCTION	FORMAT
VRENAME	Names or renames a disk	 Examples: vr 1:,nekynot vr piecharts:,stats

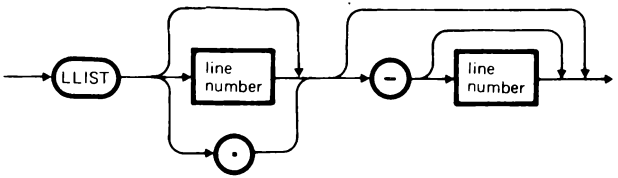
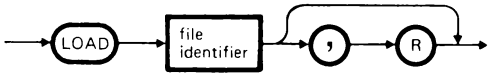
NAME	FUNCTION	FORMAT
AUTO	Starts automatic generation of line numbers	 Examples: AUTO 10,5 AUTO .,20
CONT	Restarts execution of an interrupted program	 Example: CONT

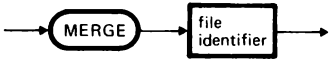
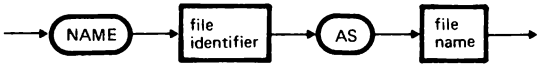
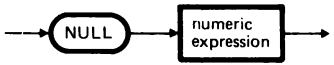
NAME	FUNCTION	FORMAT
DELETE	Deletes one or more lines of a program	

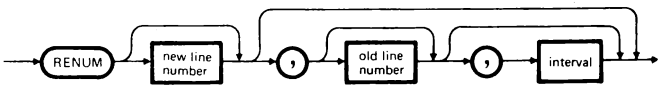
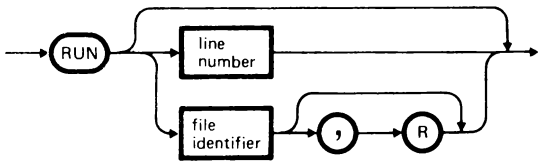
Examples: DELETE -50
DELETE 190 - 300

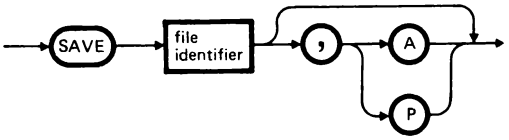
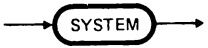
NAME	FUNCTION	FORMAT
EDIT	Allows the editing of a line of a program within Edit mode	 Example: EDIT 110 Note: See the Appendix "Edit Mode Subcommands."
FILES	Lists the names of the files in the volume directory	 Examples: FILES FILES "1:MY*"

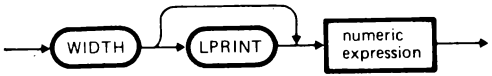
NAME	FUNCTION	FORMAT
KILL	Deletes a file from disk	 Examples: KILL "0: BUSINESS" KILL "NUMBERS/PNU"
LIST	Displays one or more lines of a program	 Examples: LIST LIST 10 - 85

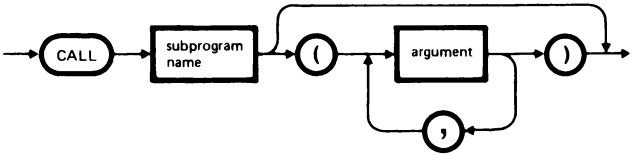
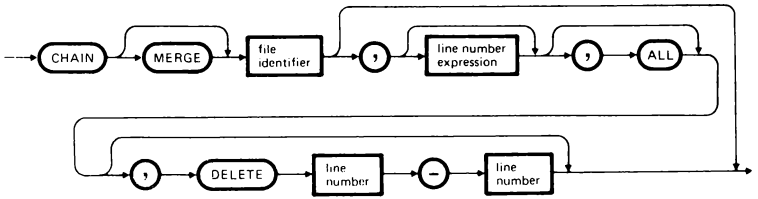
NAME	FUNCTION	FORMAT
LLIST	Prints one or more lines of a program	 Examples: LLIST LLIST 300 - 550
LOAD	Loads a program from disk and, optionally, executes it	 Examples: LOAD "INVEN" LOAD "SINES", R

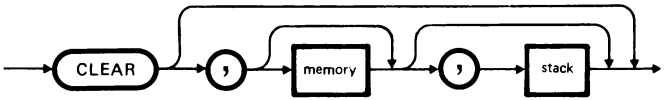
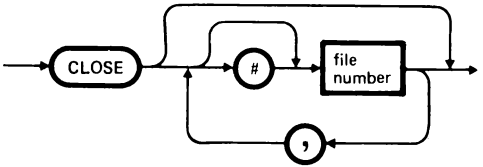
NAME	FUNCTION	FORMAT
MERGE	Merges a program on disk with one in memory	 Example: MERGE "SUBRTN"
NAME	Changes the name of a file on disk	 Example: NAME "SUB2" AS "SUB"
NEW	Deletes the program currently in memory and its variables	 Example: NEW
NULL	Sets the number of nulls printed after each line	 Example: NULL 2

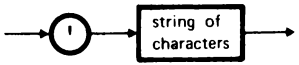
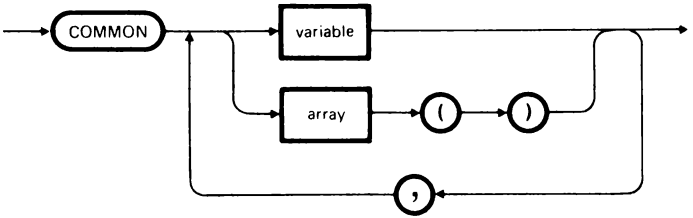
NAME	FUNCTION	FORMAT
RENUM	Changes the line numbers of a program in memory	 Examples: RENUM 10, 15, 5 RENUM
RUN	Executes all or part of a program in memory or loads a program from disk and executes it, optionally leaving open its files	 Examples: RUN A$ RUN "1:GRAPH", R

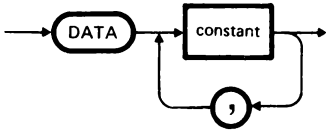
NAME	FUNCTION	FORMAT
SAVE	Saves the program in memory, copying it onto disk	 Examples: SAVE "WAHA", P SAVE "MAX", A Note: The specification of P protects the program against subsequent resaving, listing, or editing. If the program is to be merged at a later time, A must be specified.
SYSTEM	Closes all files and returns to PCOS	 Example: SYSTEM

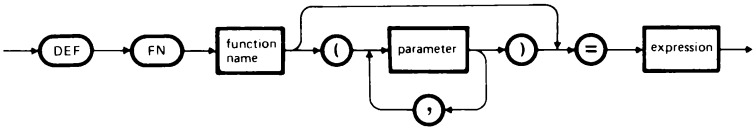
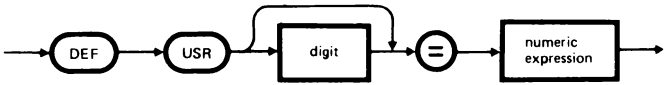
NAME	FUNCTION	FORMAT
TROFF	Stops the line number listing initiated by TRON	 Example: TROFF
TRON	Causes the line number of each statement executed to be listed	 Example: TRON
WIDTH	Establishes screen or printer width	 Example: WIDTH LPRINT 100

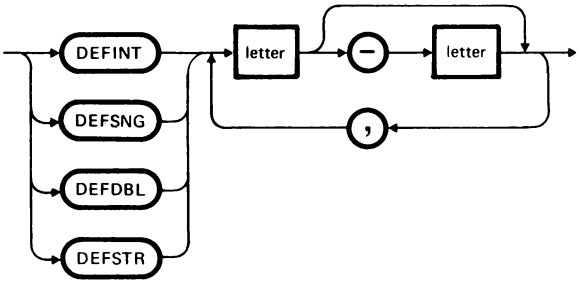
NAME	FUNCTION	FORMAT
CALL	Calls a PCOS command or Assembly language subprogram	 Examples: 50 CALL "fn" (FILE$, SIZE%) 90 CALL "SUB12" (A,B)
CHAIN	Calls a program and passes values to it	 Examples: 350 CHAIN "NEXT1" 400 CHAIN MERGE "OVERLAY", 1000

NAME	FUNCTION	FORMAT
CLEAR	Sets the value of all numeric variables to zero, of all string variables to null, and closes all open data files	 Example: 200 CLEAR
CLOSE	Closes one or more disk files	 Examples: 100 CLOSE 1, 2, 3 300 CLOSE

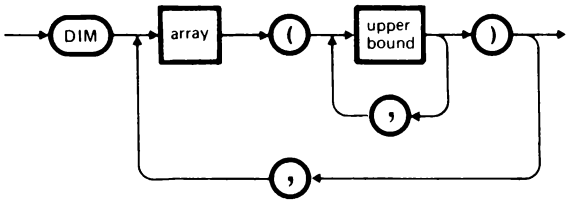
NAME	FUNCTION	FORMAT
(COMMENT FIELD)	Allows comments to be added to a statement	 Examples: 50 GOSUB 999 'GRAPHICS ROUTINE 325 GOTO 80 'VALUE NOT FOUND
COMMON	Lists the program variables to be passed to a chained program	 Example: 170 COMMON D, E, F(), T$

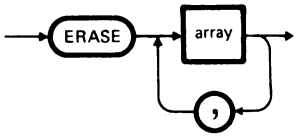
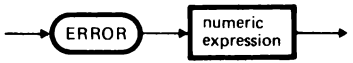
NAME	FUNCTION	FORMAT
DATA	Creates an internal data file	 Examples: 10 DATA 8, 25, 3.5, 5.55 90 DATA "NAME, FIRST", NAME, ADDRESS, 12

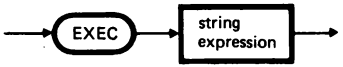
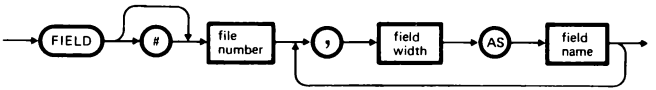
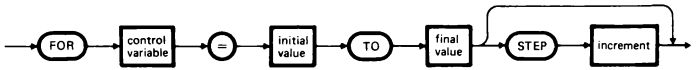
NAME	FUNCTION	FORMAT
DEF FN	Defines a numeric or string function	 10 DEF FN1(X) = (SIN(X/S)*3)/180
DEF USR	Specifies the starting address of an Assembly language subprogram	 Example: 200 DEF USR2 = 24000 Note: See USR

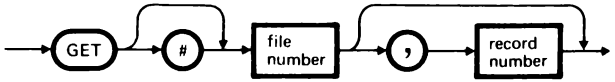
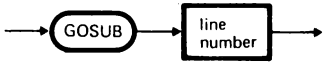
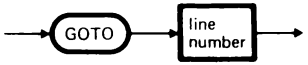
NAME	FUNCTION	FORMAT
DEFDBL	Defines double precision variables	
DEFINT	Defines integer variables	
DEFSNG	Defines single precision variables	
DEFSTR	Defines string variables	

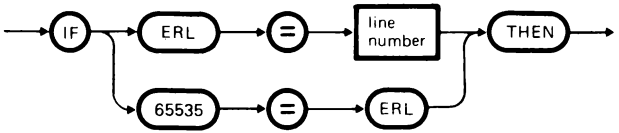
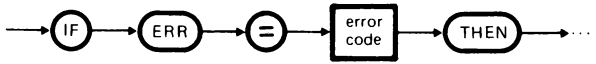
Examples:
```
10 DEFDBL M-R
10 DEFDBL D-G, J, X-Z
10 DEFINT I-N
30 DEFINT A, X-Z
20 DEFSNG S-V, X
10 DEFSNG K, E, N
10 DEFSTR A-D, S-V
10 DEFSTR T, O, N, Y
```

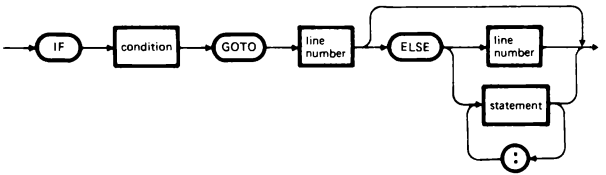
NAME	FUNCTION	FORMAT
DIM	Specifies the number of dimensions and the maximum number of elements for each dimension, for one or more arrays	 Examples: 30 DIM A(15) 50 DIM A(10,10,10), B(25,35)
END	Terminates a program, closes all files, and returns to BASIC command level	 Example: 999 END

NAME	FUNCTION	FORMAT
ERASE	Deletes arrays	 Examples: 90 ERASE A 300 ERASE C, D
ERROR	Simulates the occurrence of a BASIC error or generates a user-defined error	 Example: 60 ERROR S

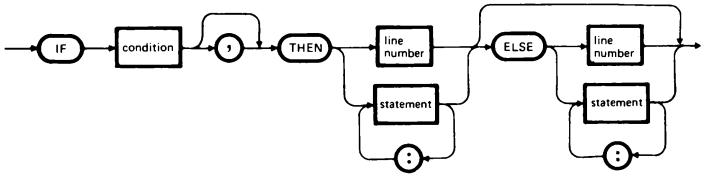
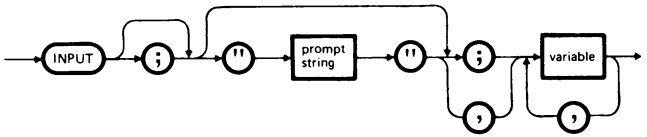
NAME	FUNCTION	FORMAT
EXEC	Calls a PCOS command or an Assembly language subprogram	 Examples: 100 EXEC "pk '#', 'RUN V1:CAS'" 240 EXEC A$
FIELD#	Allocates space for variables in a random file buffer	 Example: 70 FIELD#1, 25 AS TK$, 10 AS P$
FOR	Starts the execution of FOR/NEXT loop	 Examples: 60 FOR A% = 1 TO 10 90 FOR J% = I%*N% TO P%*R% STEP K%

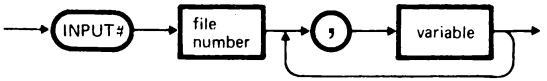
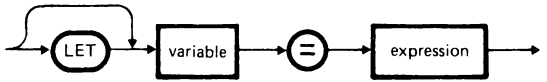
NAME	FUNCTION	FORMAT
GET#	Reads a record from a random file into a random file buffer	 Examples: 40 GET#1,3 80 GET#1
GOSUB	Transfers control to a subroutine	 Example: 65 GOSUB 999 Note: see RETURN
GOTO	Transfers control to a specified program line	 Example: GOTO 75

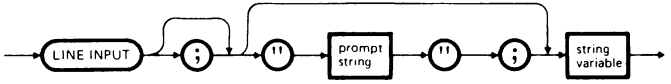
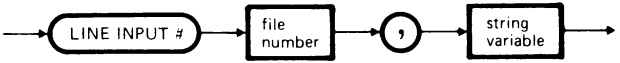
NAME	FUNCTION	FORMAT
IF...ERL...THEN	Transfers control to a specified statement after determination of the line number in which an error was detected	 Example: 80 IF ERL = 78 THEN PRINT "L=Ø"
IF...ERR...THEN	Transfers control to a specified statement based on a specified error code	 Example: 80 IF ERR = 200 THEN PRINT "W=Ø"

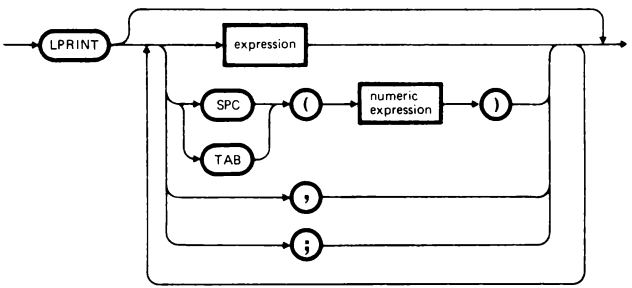
NAME	FUNCTION	FORMAT
IF...GOTO...ELSE	Transfers control, conditionally, to a specified program line	

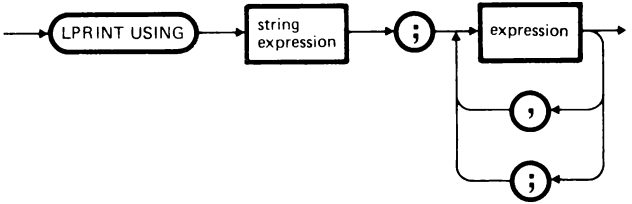
Examples: 80 IF (A=B) GOTO 90 ELSE 40
60 IF (C%=F%) OR (A>B) GOTO 150

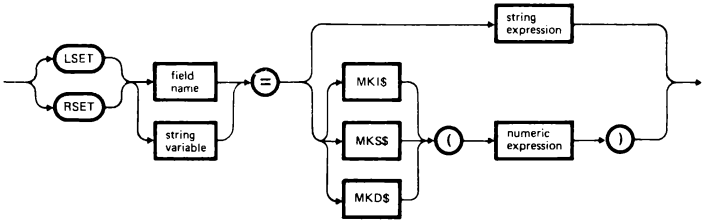
NAME	FUNCTION	FORMAT
IF...THEN...ELSE	Transfers control, conditionally, to a specified statement	 Examples: 160 IF (X>Y) THEN PRINT X; "OUT OF RANGE" 555 IF K > 88 THEN END ELSE GOTO 320
INPUT	Assigns values to variables from data entered from the keyboard	 Examples: 50 INPUT "ENTER SUM"; A 80 INPUT "ENTER NAME THEN DISTRICT"; A$, B$

NAME	FUNCTION	FORMAT
INPUT#	Assigns values to variables from a sequential file	 Examples: 60 INPUT# 1, A, B 85 INPUT# 3, S$, T$
LET	Assigns a value to a variable	 Examples: 30 A$ = "AMOUNT" 50 C = (D+E) * 10

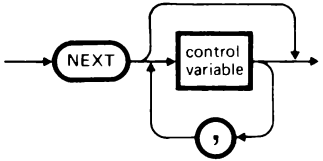
NAME	FUNCTION	FORMAT
LINE INPUT	Assigns a value to a string variable from the keyboard, without the use of delimiters	 Examples: 10 LINE INPUT "CUSTOMER?"; C$ 80 LINE INPUT "PREVIOUS ADDRESSES?"; A$
LINE INPUT#	Reads a line from a sequential file and assigns it to a string variable	 Example: 70 LINE INPUT# 1, A$

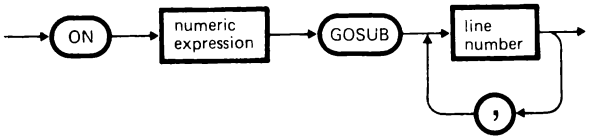
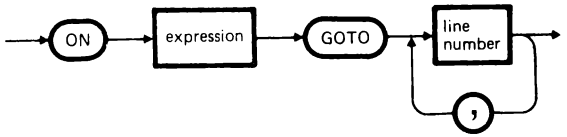
NAME	FUNCTION	FORMAT
LPRINT	Prints a list of data in standard format	 Examples: 65 LPRINT A$, B, SPC(3); C$; X+5 80 LPRINT,, A$; 376; "TONY"; "KEN"

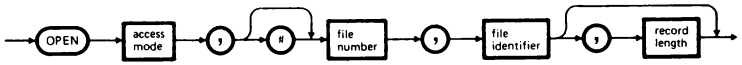
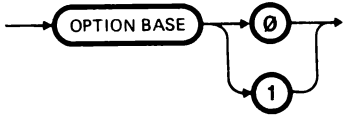
NAME	FUNCTION	FORMAT
LPRINT USING	Prints a list of data in user-defined format	 Examples: 70 LPRINT USING "##.## "; 10.2; 66.78 90 LPRINT USING "$$###.## "; A; B; C

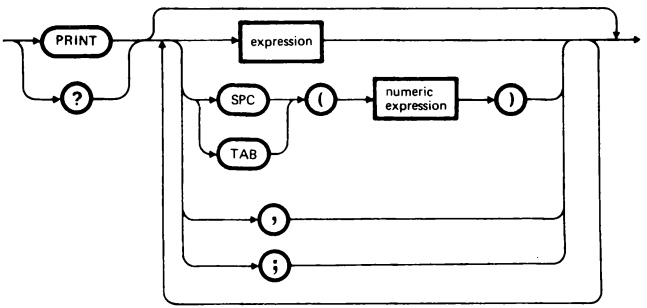
NAME	FUNCTION	FORMAT
LSET	Transfers data from memory into a random file buffer, left justifying if necessary	

Examples: 180 LSET A$ = MKS$ (AMT)
160 LSET A$ = D$ Note: see RSET

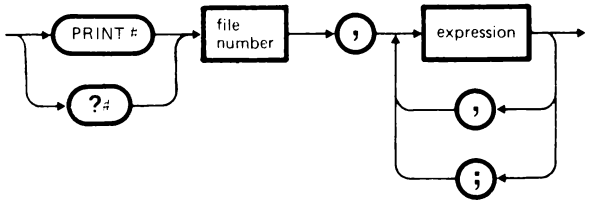
NAME	FUNCTION	FORMAT
NEXT	Defines the end of one or more FOR/NEXT loops	 Examples: 80 NEXT 350 NEXT I, J
ON ERROR GOTO	Activates error trapping and specifies the first line of the error handling routine	 Example: 120 ON ERROR GOTO 400

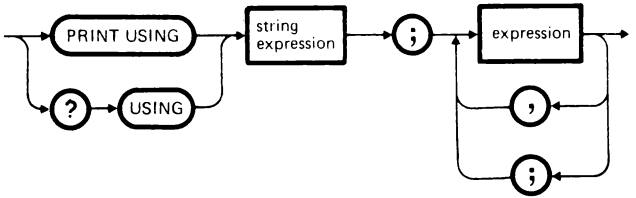
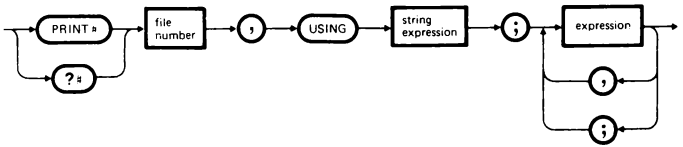
NAME	FUNCTION	FORMAT
ON...GOSUB	Transfers control to a designated subroutine based on the value of a specified expression	 Examples: 80 ON A+B GOSUB 999, 1099, 1199 45 ON X GOSUB 200, 300, 400, 500
ON...GOTO	Transfers control to a designated programs line based on the value of a specified expression	 Examples: 50 ON C GOTO 100, 200, 300 70 ON A/B GOTO 30, 75, 450

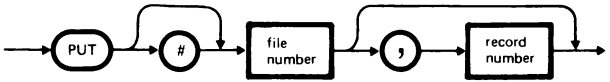
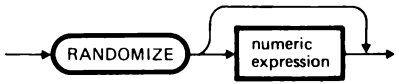
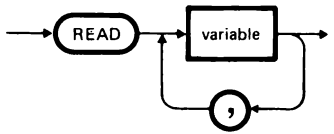
NAME	FUNCTION	FORMAT
OPEN	Opens a file stored on disk and indicates the mode in which it will be processed	 Example: 10 OPEN "I", # 2, "ACCTS"
OPTION BASE	Declares the minimum value (lower limit) for an array subscript	 Example: 10 OPTION BASE 1

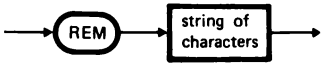
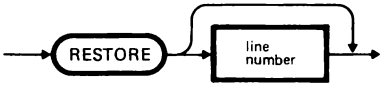
NAME	FUNCTION	FORMAT
PRINT	Displays a list of data in a standard format	

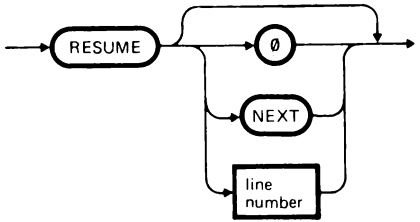

```
Examples:  60 PRINT 9.9; A; A$; SPC(3); B
           90 PRINT G$; F,, A
```

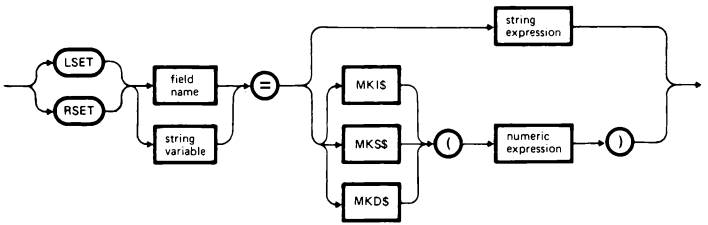
NAME	FUNCTION	FORMAT
PRINT#	Writes data into a sequential file	 Examples: 50 PRINT# 1, X; Y; Z 80 PRINT# 2, A$; ","; B$

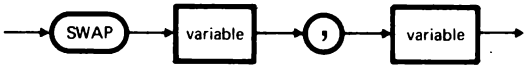
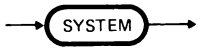
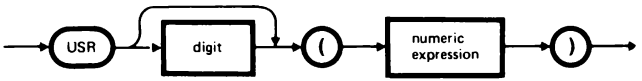
NAME	FUNCTION	FORMAT
PRINT USING	Displays a list of data in a user-defined format	 Examples: 70 PRINT USING "##.##"; 111.22 90 PRINT USING ".##"; .999
PRINT# USING	Writes data into a sequential file in user-defined format	 Example: 80 PRINT# 2,USING "$$###.##"; A; B; C

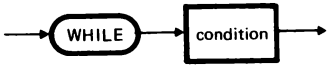
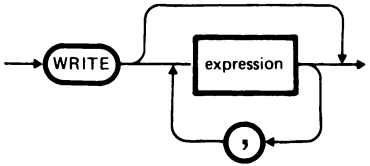
NAME	FUNCTION	FORMAT
PUT#	Writes data from a random file buffer to a random file	 Example: 40 PUT#2, K
RANDOMIZE	Generates a random number sequence different from the standard sequence	 Example: 40 RANDOMIZE 10
READ	Reads data from an internal data file	 Example: 30 READ A, B, X$

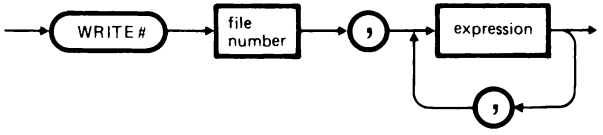
NAME	FUNCTION	FORMAT
REM	Allows explanatory remarks to be inserted in a program	 Example: 60 REM PRIME NUMBER GENERATOR
RESTORE	Moves the pointer either to the beginning of an internal data file or to a specified line number	 Example: 70 RESTORE

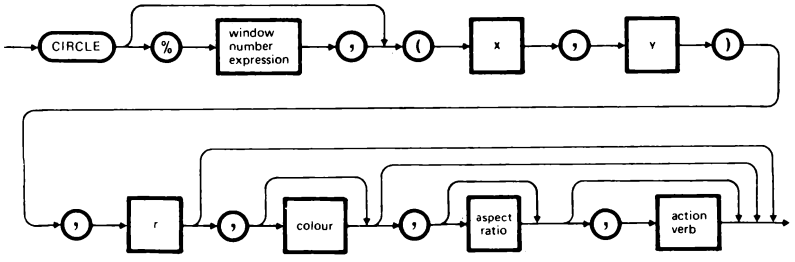
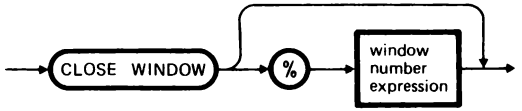
NAME	FUNCTION	FORMAT
RESUME	Resumes execution after an error handling routine has been entered	 Example: 150 RESUME
RETURN	Returns control to the statement following a GOSUB	 Example: 140 RETURN

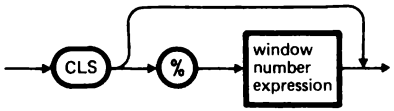
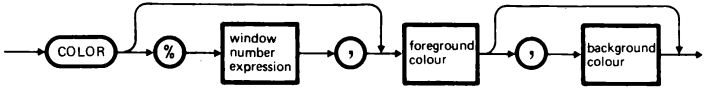
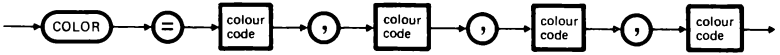
NAME	FUNCTION	FORMAT
RSET	Transfers data from memory into a random file buffer, right justifying if necessary	 Example: 120 RSET C$ = MKS$ (NUM) 150 RSET A$ = F$ Note: see LSET
STOP	Interrupts program execution and returns to command mode	 Example: 120 STOP

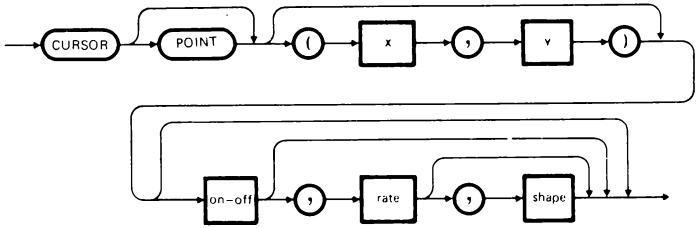
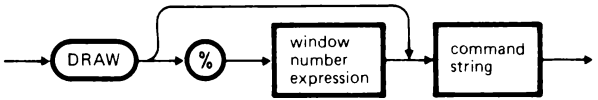
NAME	FUNCTION	FORMAT
SWAP	Exchanges the values of two variables	 Examples: 70 SWAP A, B 90 SWAP A$, B$
SYSTEM	Returns to PCOS, clears memory, and closes all data files	 Example: 220 SYSTEM
USR	Calls an Assembly language subprogram, passing one argument to it	 Example: 50 D = USR (B/3) Note: see DEF USR

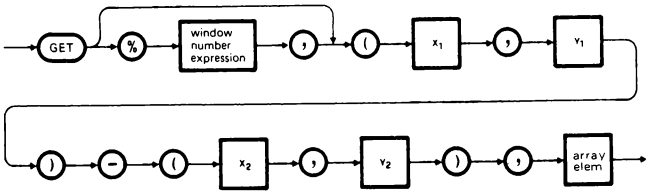
NAME	FUNCTION	FORMAT
WEND	Defines the end of WHILE/WEND loop	 Example: 180 WEND
WHILE	Starts the execution of a WHILE/WEND loop	 Examples: 75 WHILE A-B 90 WHILE C
WRITE	Displays a list of data	 Example: 60 WRITE A, B, C$

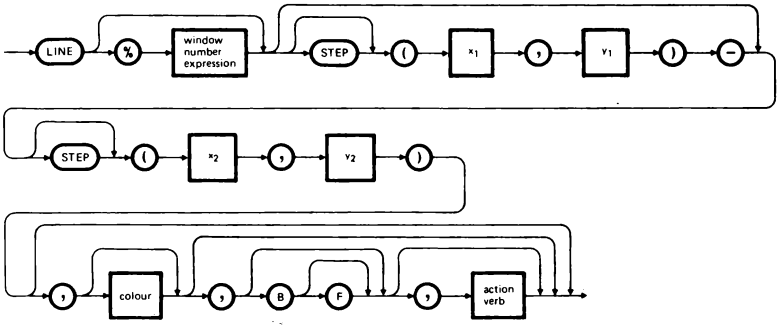
NAME	FUNCTION	FORMAT
WRITE#	Writes data into a sequential file adding delimiters between values	 Examples: 80 WRITE#1, A, B 90 WRITE#2, A$, B$

NAME	FUNCTION	FORMAT
CIRCLE	Draws either a circle or an ellipse	 Example: 10 CIRCLE (50,50), 20
CLOSE WINDOW	Closes a window	 Example: CLOSE WINDOW %B

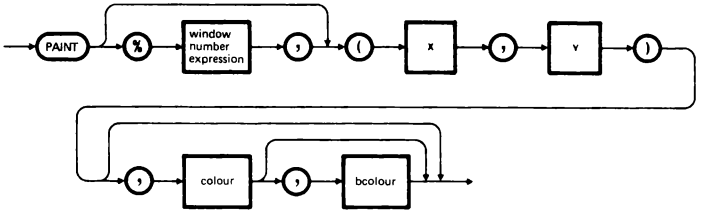
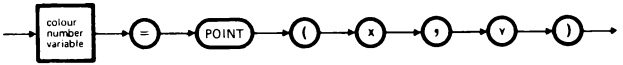
NAME	FUNCTION	FORMAT
CLS	Cancels the contents of a window.	 Example: 170 CLS %A
COLOR (for a window)	Chooses the background and foreground colors for a selected window.	 Example: 60 COLOR %A, (0,1)
COLOR (global colorset selection)	Chooses the colors to work with on the color display	 Example: 40 COLOR = 3, 5, 6, 7 Note: 0 = black; 1 = green; 2 = blue; 3 = cyan; 4 = red; 5 = yellow; 6 = magenta; 7 = white

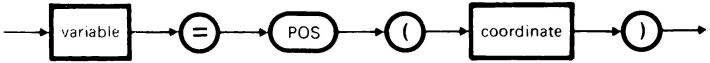
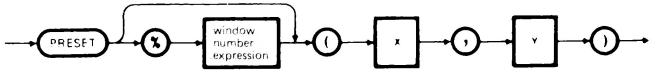
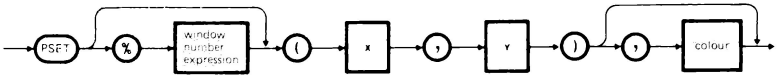
NAME	FUNCTION	FORMAT
CURSOR	Specifies the position of the cursors, determines their shape, whether they are to be displayed, and whether they are to blink	 Example: 60 CURSOR POINT (50,50) 1,1
DRAW	Moves the graphic cursor within a window and draws lines in a given color	 Example: 140 DRAW "M = X, 25"

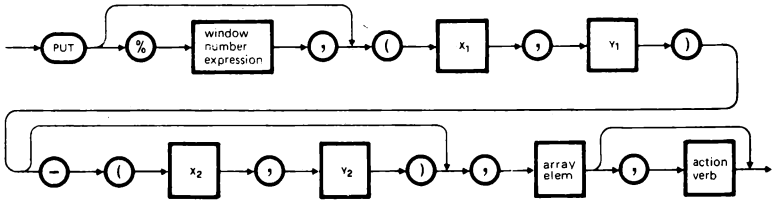
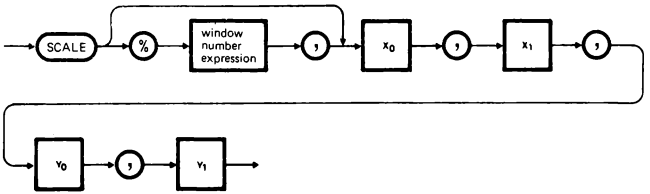
NAME	FUNCTION	FORMAT
GET	Saves the whole or a part of a window in a specified array	 Example: 150 GET (5,7) - (10,9), A

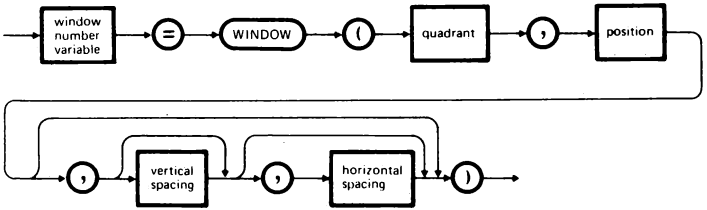
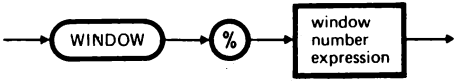
NAME	FUNCTION	FORMAT
LINE	Draws either a line or a rectangle and fills it with a specified color	

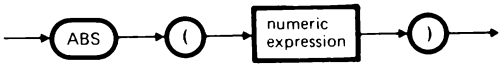
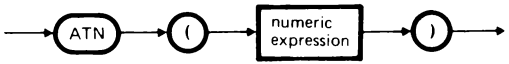
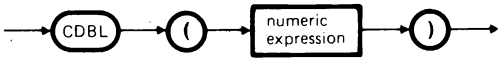
Example: 70 LINE (10,70) - (155,90),,B

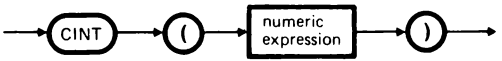
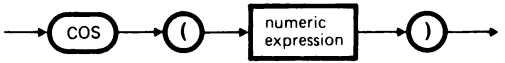
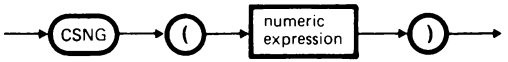
NAME	FUNCTION	FORMAT
PAINT	Colors the whole or a portion of a window starting from the pixel closest to the x,y coordinates	 Example: 20 PAINT (50,50), 1,1
POINT	Returns the color code of the pixel nearest to the specified coordinates	 Example: 60 A% = POINT (50,50)

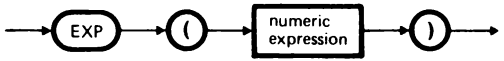
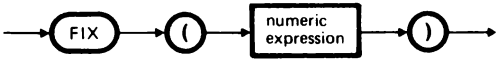
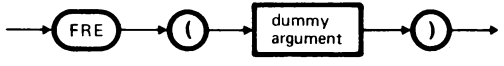
NAME	FUNCTION	FORMAT
POS	Returns the position of the text cursor in the current window	 Example: 70 B = POS (0)
PRESET	Colors the pixel closest to the x,y coordinates with the background color of either the current or selected window	 Example: 50 PRESET %A (10,20)
PSET	Colors the pixel closest to the specified coordinate with a specified color	 Example: 80 PSET (10,23), 2

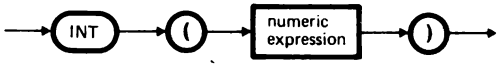
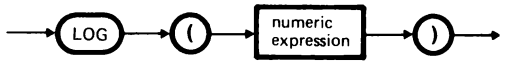
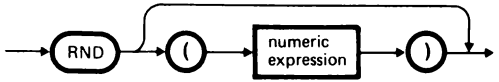
NAME	FUNCTION	FORMAT
PUT	Puts a window or part of a window onto the display from a specified array	 Example: 80 PUT (7,15) - (9,25), B
SCALE	Defines the scaling between the user's "problem" coordinates and the "hardware" coordinates	 Example: 70 SCALE -1000, 1000, -1000, 1000

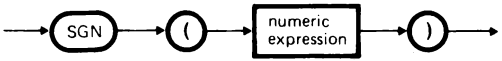
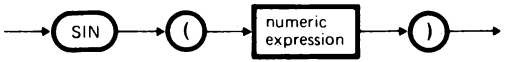
NAME	FUNCTION	FORMAT
WINDOW (Format 1)	Opens a new window by splitting the current windows, or with quadrant and position specified as 0 returns the number of the current window	 Examples: 50 A = WINDOW (0,100,14) 70 B = WINDOW (0,0)
WINDOW (Format 2)	Selects a windpw as the current window	 Example: 90 WINDOW %A

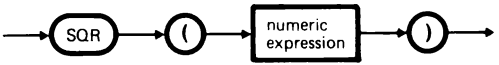
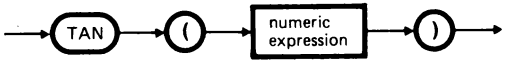
NAME	VALUE RETURNED	FORMAT
ABS	Absolute value of an algebraic number	 Examples: 40 LET A = B - ABS(C/D) 60 ? ABS(D/C)
ATN	Arctangent of an angle (in radians)	 Examples: 20 PRINT ATN (45.78) 90 X = ATN (Y+Z)
CDBL	Double precision value of a number	 Example: 20 A = CDBL (Y) 50 LPRINT CDBL (A*B)

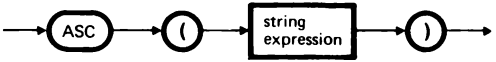
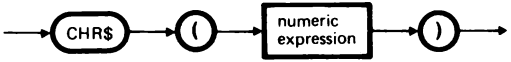
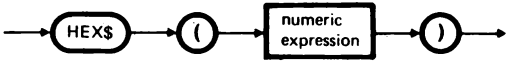
NAME	VALUE RETURNED	FORMAT
CINT	Integer part of a number (rounded)	 Example: 80 B = CINT (B)
COS	Cosine of an angle (in radians)	 Examples: 50 PRINT COS (A+B) 90 A = COS (34)
CSNG	Single precision value of a number	 Examples: 70 C = CSNG (X#) 95 C = CSNG (X-Y)

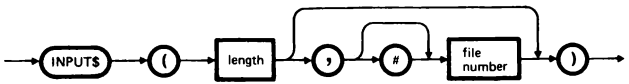
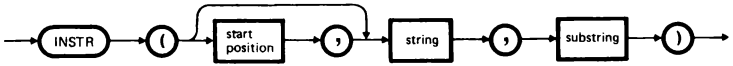
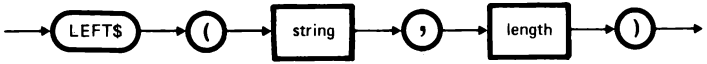
NAME	VALUE RETURNED	FORMAT
EXP	Natural exponential of a number	 Examples: 70 PRINT EXP (X-3) 90 PRINT EXP (5)
FIX	Integer part of a number	 Examples: 30 B = FIX (B) 60 X = FIX (X/Y)
FRE	Number of bytes in memory not used by BASIC	 Examples: PRINT FRE (X$) PRINT FRE (" ")

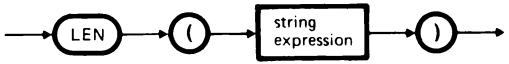
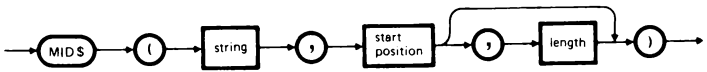
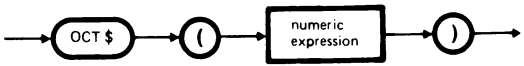
NAME	VALUE RETURNED	FORMAT
INT	Greatest integer less than or equal to a number	 Examples: 130 A = INT (C/B) 45 PRINT INT (X)
LOG	Natural logorithm of a number	 Examples: 60 A = LOG (26.7) 70 PRINT A + LOG (B)
RND	Random number between 0 and 1	 Examples: 40 X = RND (1) 90 PRINT INT (RND*50) Note: If num-exp < 0, same sequence is restarted; if num-exp = 0, last random

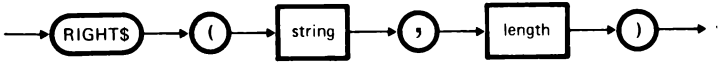
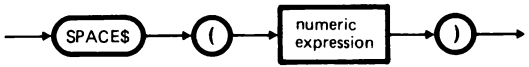
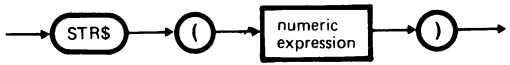
NAME	VALUE RETURNED	FORMAT
		number generated is repeated; if num-exp > 0 or omitted next random number is generated
SGN	Indication of the sign of a number	 Examples: 95 ON SGN(X) + 2 GOTO 100, 200, 300 60 PRINT SGN (Y+3) Note: If num-exp < 0, a -1 is returned; If num-exp = 0, a 0 is returned; If num-exp > 0, a 1 is returned
SIN	Sine of an angle (in radians)	 Examples: 80 PRINT SIN (V) 60 B = SIN (56)

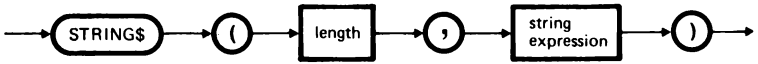
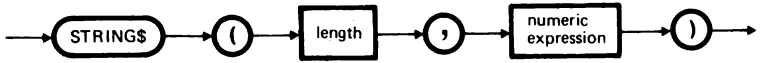
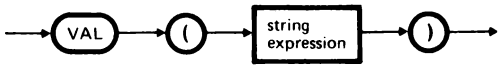
NAME	VALUE RETURNED	FORMAT
SQR	Square root of a positive number	 Examples: 30 A = SQR (255)*C 70 PRINT SQR (D)
TAN	Tangent of an angle (in radians)	 Examples: 60 PRINT TAN (23) 95 A = TAN (D+G)

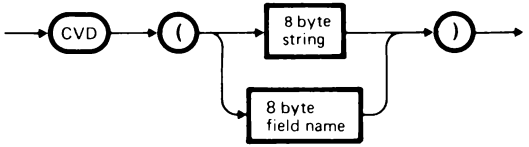
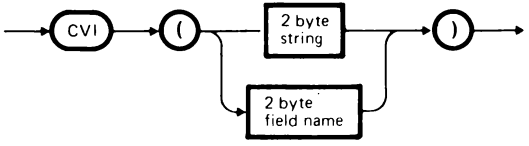
NAME	VALUE RETURNED	FORMAT
ASC	ASCII value of the first character of a string	 Example: 40 PRINT ASC (A$)
CHR$	ASCII character corresponding to a numeric code	 Example: 60 PRINT CHR$ (84)
HEX$	String value representing the hexadecimal value of a decimal argument	 Example: 50 A$ = HEX$ (X)
INKEY$	One-character string corresponding to a key pressed at keyboard	 Example: 999 A$ = INKEY$

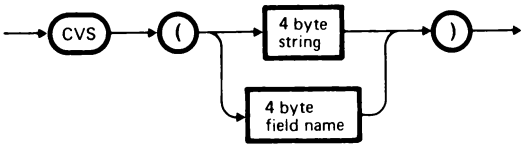
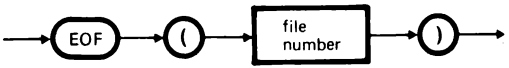
NAME	VALUE RETURNED	FORMAT
INPUT$	String of characters read from the keyboard or a file	 Example: 80 X$ = INPUT$ (1)
INSTR	Position in a string of the first occurrence of a substring, starting from a specified position	 Examples: 95 PRINT INSTR (4, "ABCDEBF", "B") 40 I% = INSTR (A$ + B$, C$)
LEFT$	Substring of a specified length extracted starting from the leftmost position of a string	 Examples: 50 B$ = LEFT$ (A$,5) 70 C$ = LEFT$ (C$+"BAT",7)

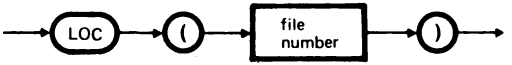
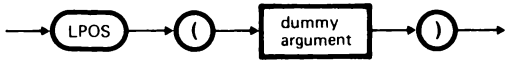
NAME	VALUE RETURNED	FORMAT
LEN	Number of characters in a string	 Example: 130 A = LEN (B$) + LEN(C$)
MID$	Substring of a specified length extracted starting from a specified position	 Example: 60 PRINT MID$ (B$,9,7)
OCT$	String value representing the octal value of a decimal argument	 Example: 40 N$ = OCT$ (A+B)

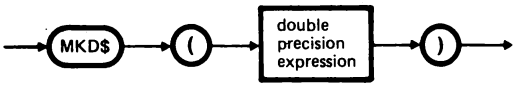
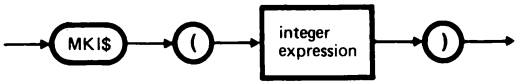
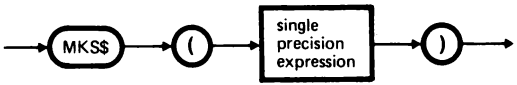
NAME	VALUE RETURNED	FORMAT
RIGHT$	Substring of a specified length extracted starting from the righmost position of a string	 Example: 70 A$ = RIGHT$ (B$,8)
SPACE$	String consisting of a specified number of spaces	 Example: 80 B$ = SPACE$ (I)
STR$	String representation of a numeric value	 Examples: 30 T$ = STR$(N) 80 A$ = STR$(3.14)

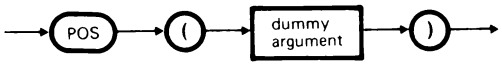
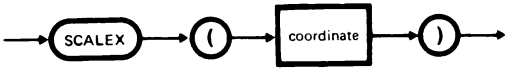
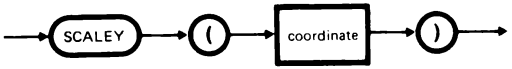
NAME	VALUE RETURNED	FORMAT
STRING$	Substring of a specified length each character of which is the first character of a specified string	 Examples: 70 A$ = STRING$ (15,X$) 90 A$ = STRING$ (N,"_")
STRING$	String of a specified length each character of which is the representation of a specified ASCII numeric code	 Examples. 50 X$ = STRING$ (10,45) 80 X$ = STRING$ (A,Y)
VAL	Numeric value of a string expression	 Example: 80 Y = VAL(A$)

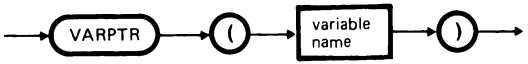
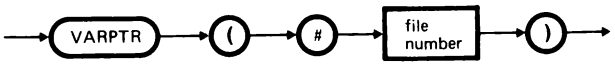
NAME	VALUE RETURNED	FORMAT
CVD	Double precision number derived from conversion of an 8-character string	 Example: 50 C# = CVD(X$)
CVI	Integer derived from conversion of a 2-character string	 Example: 40 A% = CVI(C$)

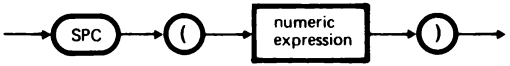
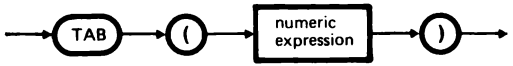
NAME	VALUE RETURNED	FORMAT
CVS	Single precision number derived from conversion of a 4-character string	 Example: 20 Y! = CVS(N$)
EOF	Indication (-1) that end of a sequential file has been reached	 Example: 90 IF EOF(3) THEN 120
ERL	Line number of a line in which an error was detected	 Example: 900 PRINT "Too big", ERL Note: See IF...ERL...THEN

NAME	VALUE RETURNED	FORMAT
ERR	Error code of an error detected during execution	 Example: 200 I%=ERR Note: See IF...ERR...THEN
LOC	Next record number to read or write (random file) or number of sectors read or written (sequential file)	 Examples: 60 PRINT LOC(1) 80 PRINT LOC(2)
LPOS	Current position of the line printer print head within the line printer buffer	 Example: 80 IF LPOS(X)>60 THEN LPRINT A$

NAME	VALUE RETURNED	FORMAT
MKD$	An 8-character string derived from the conversion of a double precision value	 Example: 70 LSET A$ = MKD$(C#)
MKI$	A 2-character string derived from the conversion of an integer value	 Example: 60 LSET N$ = MKI$(A%)
MKS$	A 4-character string derived from the conversion of a single precision value	 Example: 60 LSET F$ = MKS$(A)

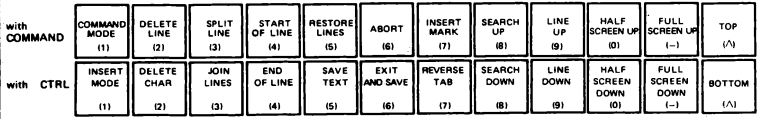
NAME	VALUE RETURNED	FORMAT
POS	Current character position of the cursor	 Example: 70 IF POS(X) > 60 THEN PRINT A$
SCALEX	Pixel coordinate on the x-axis corresponding to a specified user cordinate	 Example: 80 PRINT SCALEX (.25)
SCALEY	Pixel coordinate on the y-axis corresponding to a specified user coordinate	 Example: 60 PRINT SCALEY (.75)

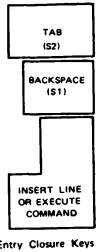
NAME	VALUED RETURNED	FORMAT
VARPTR	Address of variable in memory or zero if variable has not been assigned a value	 Example: 40 I = VARPTR (X)
VARPTR	For sequential files, starting address of the disk I/O buffer assigned to the specified file; for random files, address of the FIELD buffer associated with the specified file	 Example: 60 J = VARPTR (#2)

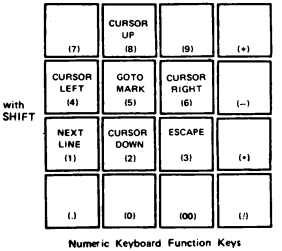
NAME	PURPOSE	FORMAT
SPC	To specify the number of spaces to precede or follow the printing or display of a data item	 Examples: 40 PRINT B$; SPC(10); C$ 85 LPRINT A; SPC(25); "YEAR"
TAB	To specify the position in a line at which a data item is to be printed or displayed	 Examples: 60 LPRINT A, TAB(35), B$ 80 PRINT TAB(I), "___"

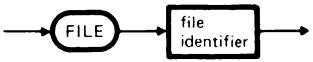
VIDEO FILE EDITOR

The video file editor is invoked, from PCOS, via the EDIT command. Once EDIT has been issued, the keyboard operates as indicated in the following diagrams. The characters shown on the "Top Row Function Keys" --(1),(2), ...(^) -- are those that appear on the USA ASCII keyboard. Positionally, though, the functions provided by these keys are the same on all keyboards; it is never necessary to press SHIFT with COMMAND or CTRL to achieve a desired effect.







NAME	FUNCTION	FORMAT
DELETE	Removes all text between the current line and the MARK line, placing the removed text in the restore buffer	 Example: delete
FILE	Suspends processing of the current file and invokes the editor for processing of another file	 Example: file alys/arty
GOTO	Moves the window to a specified line number in the file	 Example: goto 7

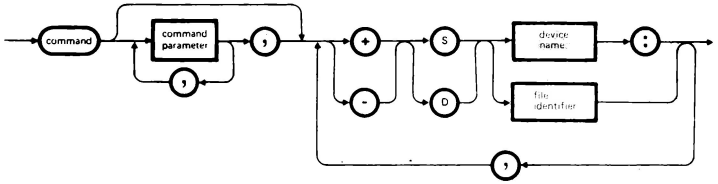
Note: Before entering a command, you must strike the COMMAND MODE function key. To execute the command, you strike the EXECUTE COMMAND key.

DEVICE RE-ROUTING

Information relating to device re-routing is provided below.

LOCAL DEVICE RE-ROUTING

Re-routes input and output of the PCOS command being executed.



Note: Default device names are:

prt: = PCOS Printer Driver
cons: = PCOS Console Driver (video and keyboard)
ieee: = IEEE 488 Driver
Com: = Standard RS 232-C communication port
Com1: = First RS 232-C communication port on TWIN BOARD
Com2: = Second RS 232-C communication port on TWIN BOARD

Note that + or -, S or D and device name or file name follow each other without any spaces in between, and constitutes one device re-routing parameter. Each device re-routing parameter must be followed by a comma to separate it from the next device re-routing parameter. Note also that +drpt: may also be written as +prt (in capital or small letters).

Note:

Adding simply +PRT to a command -- for example:

```
ba +prt
```

causes that command's output to be directed to the printer as well as to the video screen. (As a result, all BASIC activities will appear on the printer and the screen.)

Examples:

```
vl 1: -dcons: +dprt:
```

causes the output of the VLIST command, directed to the disk in drive 1, to be printed only.

```
vl +prt
```

causes the output of the VLIST command, directed to the disk in the last drive referred to, to be printed as well as displayed.

GLOBAL DEVICE RE-ROUTING

Re-routes input and output of all subsequent PCOS commands executed.

To activate global device re-routing, specify the parameters shown in the syntax diagram for local device re-routing, omitting COMMAND and COMMAND PARAMETERS.

Examples:

+prt or +dprt:

causes all subsequent output to be printed as well as displayed.

-dcons: +dprt:

causes all subsequent output to be printed only.

APPENDIX A EDIT MODE SUBCOMMANDS

An EDIT command must be issued before any of the following subcommands can be used.

SUBCOMMAND	FUNCTION
A	Restores the line as it was and restarts editing from the start of the line
CR	Displays the newly modified line, exits from edit mode, and returns to command mode
n C characters	Changes the next n characters in the line to the characters entered from the keyboard. (The number of characters entered must equal the value of n.)
n D	Deletes n characters starting from the position of the cursor
E	Ends editing, returns to Command mode, saves changes, but does not display the rest of the line
H characters	Deletes the rest of the line, starting from the position of the cursor, and allows the optional insertion of characters at that point
I characters	Inserts the characters keyed in on the keyboard at the position of the cursor. (To end insert state, press CTRL HOME to continue editing the line, or CR to exit from edit mode.)
n K character	Kills all characters up to the nth occurrence of Character
L	Lists the rest of the line, from the position of the cursor and returns to the start of the line

Q	Restores the line as it was and exits from edit mode into command mode
n S character	Searches for the nth occurrence of character and positions the cursor in front of it
X characters	Inserts characters at the end of the line

APPENDIX B USE OF THE CONTROL KEY CTRL

The Control Key CTRL is pressed together with other keys to perform a number of special operations. The correspondence between the keys pressed and the operation performed is shown below.

KEYS PRESSED	OPERATION PERFORMED
CTRL C	Activates the Break facility to delete the line being typed. Interrupts program execution or automatic line numbering and returns the M20 to BASIC command mode
CTRL G	Suppresses the display or printing of the characters about to be entered
CTRL H	Deletes the last character entered
CTRL I	Provides a tab of eight spaces
CTRL S	Suspends program execution or listing (to resume, press any key.)
CTRL U	Deletes the line being typed
CTRL HOME	Exits insert state without exiting edit mode
CTRL RESET	Clears memory and reloads PCOS and BASIC from disk

APPENDIX C ERROR MESSAGES AND CODES

ERROR CODE	MESSAGE	EXPLANATION
1	NEXT WITHOUT FOR	A NEXT statement has been encountered without a matching FOR.
2	SYNTAX ERROR	A line has been encountered which includes an incorrect sequence of characters (misspelled keyword, incorrect punctuation, etc.).
3	RETURN WITHOUT GOSUB	A RETURN has been encountered for which there is no previous unmatched GOSUB statement.
4	OUT OF DATA	A READ statement has been executed when there are no DATA statements with unread data remaining in the program.
5	ILLEGAL FUNCTION CALL	A parameter that is out of range has been passed to a mathematical or string function. This error may also occur with invalid arguments in function parameters or an ON...GOTO, and when: a. An array subscript is either negative or too large. b. A log function is assigned a negative or a null argument.

		c. The SQR function is assigned a negative value. d. A negative value has a non-integer exponent. e. A USR function has been called without having an initial address established.
6	OVERFLOW	The result of a calculation is too large to be represented in BASIC's number format. Note: With underflow, the result is taken as zero, and execution continues without indication of an error.
7	OUT OF MEMORY	A program is too big; or has too many loops, GOSUBs or variables; or has expressions too complicated to evaluate.
8	UNDEFINED LINE NUMBER	A line reference is to a non-existent line from a GOTO, GOSUB, IF..THEN..ELSE or DELETE statement.
9	SUBSCRIPT OUT OF RANGE	An array element has been referred to either with a subscript that is outside the dimensions of the array or with the wrong number of subscripts.
10	DUPLICATE DEFINITION	Two DIM statements have been given for the same array, or a DIM statement has also been ap-

		plied to an array after the default dimension of 10 was previously established for that array.
11	DIVISION BY ZERO	A division by zero has been encountered or the value zero has been raised to a negative power. In the former case, the result is machine infinity (with the appropriate sign); in the latter, the result is positive machine infinity.
12	ILLEGAL DIRECT	A statement which is invalid in immediate (direct) mode has been entered as an immediate command.
13	TYPE MISMATCH	A string variable name has been assigned a numeric value or vice-versa; a function that expects a numeric argument has been given a string argument or vice versa.
14	OUT OF STRING SPACE	String variables have caused BASIC to exceed the amount of free user memory remaining. (BASIC will allocate space dynamically until it runs out of memory.)
15	STRING TOO LONG	An attempt has been made to create a string more than 255 characters long.

16	STRING FORMULA TOO COMPLEX	A string expression is too long or too complex to be processed. It should be broken into smaller expressions.
17	CAN'T CONTINUE	An attempt has been made to continue a program that is noncontinuable; as it is halted due to an error, was modified during a break in execution, or does not exist.
18	UNDEFINED USER FUNCTION	A function has been called that has not been previously defined.
19	NO RESUME	An error-trapping routine has been entered that contains no RESUME statement.
2Ø	RESUME WITHOUT ERROR	A RESUME statement has been encountered before an error-trapping routine is entered.
21	UNPRINTABLE ERROR	An error message is not printable i.e. corresponds to an error with an undefined error code.
22	MISSING OPERAND	An expression contains an operator but no following operand.
23	LINE BUFFER OVERFLOW	An attempt has been made to enter a line with more than 255 characters.
24	Unprintable error.	
25	Unprintable error.	

26	FOR WITHOUT NEXT	A FOR has been encountered with out a matching NEXT.
29	WHILE WITHOUT WEND	A WHILE has been encountered without a matching WEND.
30	WEND WITHOUT WHILE	A WEND has been encountered without a matching WHILE.
31	IEEE INVALID TALKER/LISTENER ADDRESS	Use of invalid talker listener address.
32	IEEE: TALKER = LISTENER ADDRESS	An attempt has been made to talk to a talker, or listen to a listener.
33	IEEE: UNPRINTABLE ERROR	An error message is not printable i.e. corresponds to an error with an undefined error code.
34	IEEE: BOARD NOT PRESENT	An attempt has been made to use IEEE on a machine which does not have the optional IEEE interface.
35	WINDOW NOT OPEN	An attempt has been made to use a window which is not currently open.
36	UNABLE TO CREATE WINDOW	The window to be created is too big or too small for its mode (graphics or text).
37	INVALID ACTION VERB	An action verb has been incorrectly spelled or used.
38	PARAMETER OUT OF RANGE	One or more parameters have exceeded the limits set for their range.

39	TOO MANY DIMENSIONS	An attempt has been made to use an array of more than one dimension in graphics mode.
50	FIELD OVERFLOW	A FIELD statement has attempted to allocate more bytes than were specified for the record length of a random file.
51	INTERNAL ERROR	An internal malfunction has occurred. Report the conditions under which the error occurred to your Support Organization.
52	BAD FILE NUMBER	A statement or command refers to a file having a file number not within the range specified at initalization or the corresponding file is not open.
53	FILE NOT FOUND	A LOAD, KILL or OPEN statement refers to a file that does not exist on the current disk.
54	BAD FILE MODE	An attempt has been made to use random file operations (GET or PUT) with a sequential file; to use the sequential operation LOAD with a random file; or to use an invalid file mode with OPEN, i.e., not A,I,O, or R.
55	FILE ALREADY OPEN	A sequential OPEN,O, has been issued for a file that is already open, or a KILL has been applied to a file that is open.

57	DISK I/O ERROR	An input/output error has occured during a disk I/O operation.
58	FILE ALREADY EXISTS	The filename specified is identical to a filename already in use on the disk.
61	DISK FULL	All disk storage space available is in use.
62	INPUT PAST END	An INPUT statement has been executed after all the data has been assigned, or for an empty (null) file. Note: The EOF function can be used to detect end of file.
63	BAD RECORD NUMBER	The record number used with a GET or PUT statement exceeds range, i.e. is Ø or greater than 32767.
64	BAD FILE NAME	An invalid form of filename has been used with KILL, LOAD, OPEN or SAVE e.g.: - too long - includes invalid characters such as space or hyphen.
66	DIRECT STATEMENT IN FILE	A direct (immediate) statement has been encountered when loading an ASCII format file. The LOAD operation is terminated.

67	TOO MANY FILES	An attempt has been made to create a new file (using SAVE or OPEN) when the present directory is already full.
69	VOLUME NAME NOT FOUND	The volume name specified is not the name of any volume currently loaded.
70	RENAME ERROR	An error has been made during a rename operation.
71	VOLUME NUMBER ERROR	An error has been made in specifying the volume number.
72	VOLUME NOT ENABLED	A required password has not been specified.
	or	
	FILE RENAME ERRR	An error has occurred during a rename operation.
73	INVALID PASSWORD	An invalid password has been specified.
	or	
	INVALID VOLUME NUMBER	The volume number specified is invalid.
75	WRITE PROTECTED	An attempt has been made to write to a file that is write-protected.
	or	
	VOLUME NOT ENABLED	The required volume password has not been specified.
76	ERROR IN PARAMETER	The parameter specified contains an error.
	or	
	PASSWORD NOT VALID	An incorrect password has been specified.

77	ILLEGAL DISK CHANGED or TOO MANY PARAMETERS	Disk changed without closing files. The number of parameters specified exceeds the number required.
78	FILE NOT OPEN	An attempt has been made to write to or read from a file that is not yet open.
9Ø	ERROR IN PARAMETER	One or more of the quoted parameters contains an unacceptable value.
91	TOO MANY PARAMETERS	Too many parameters have been specified.
92	COMMAND NOT FOUND	Use of invalid keyword or command not available as system software is not loaded.
96	FILE NOT OPEN	The operation requested requires the specified file to be open.
97	READ FILE ERROR	An error has been encountered in reading from a file.
98	BAD LOAD FILE IDENTIFIER	An incorrect file identifier has been specified.
99	BAD LOAD FILE VERSION NUMBER	The specified version number is incorrect.
1ØØ	COMMAND ALREADY RESIDENT	An attempt has been made to pload a resident command.
1Ø1	BAD DATE OPTION	An invalid date has been specified.

1Ø2	BAD TIME OPTION	An invalid time has been specified.
1Ø3	GET DATE ERROR	An error has been encountered in a date operation.
1Ø4	GET TIME ERROR	An error has been encountered in a time operation.
1Ø5	BAD VERIFY OPTION	An invalid verify has been requested.
1Ø6	FUNCTION KEY ALREADY DEFINED	An attempt has been made to pkey a function to a key which has already been allotted a function.
1Ø8	ERROR IN CALL USER INTERFACE	An error has been encountered in a call user interface.
1Ø9	PRINTER NOT CONNECTED	The requested operation requires the connection of a printer.
111	INVALID DEVICE NAME	An invalid device name has been specified.
112	INVALID DEVICE TYPE	An invalid device type has been specified.

APPENDIX D ASCII CHARACTER CODE

This table shows decimal, hexadecimal, and binary representation of the ASCII code.

a	b	c	d	a	b	c	d	a	b	c	a	b	c
0	00	0000 0000	NUL	64	40	0100 0000	@	128	80	1000 0000	192	C0	1100 0000
1	01	0000 0001	SOH	65	41	0100 0001	A	129	81	1000 0001	193	C1	1100 0001
2	02	0000 0010	STX	66	42	0100 0010	B	130	82	1000 0010	194	C2	1100 0010
3	03	0000 0011	ETX	67	43	0100 0011	C	131	83	1000 0011	195	C3	1100 0011
4	04	0000 0100	EOT	68	44	0100 0100	D	132	84	1000 0100	196	C4	1100 0100
5	05	0000 0101	ENQ	69	45	0100 0101	E	133	85	1000 0101	197	C5	1100 0101
6	06	0000 0110	ACK	70	46	0100 0110	F	134	86	1000 0110	198	C6	1100 0110
7	07	0000 0111	BEL	71	47	0100 0111	G	135	87	1000 0111	199	C7	1100 0111
8	08	0000 1000	BS	72	48	0100 1000	H	136	88	1000 1000	200	C8	1100 1000
9	09	0000 1001	HT	73	49	0100 1001	I	137	89	1000 1001	201	C9	1100 1001
10	0A	0000 1010	LF	74	4A	0100 1010	J	138	8A	1000 1010	202	CA	1100 1010
11	0B	0000 1011	VT	75	4B	0100 1011	K	139	8B	1000 1011	203	CB	1100 1011
12	0C	0000 1100	FF	76	4C	0100 1100	L	140	8C	1000 1100	204	CC	1100 1100
13	0D	0000 1101	CR	77	4D	0100 1101	M	141	8D	1000 1101	205	CD	1100 1101
14	0E	0000 1110	SO	78	4E	0100 1110	N	142	8E	1000 1110	206	CE	1100 1110
15	0F	0000 1111	SI	79	4F	0100 1111	O	143	8F	1000 1111	207	CF	1100 1111
16	10	0001 0000	DLE	80	50	0101 0000	P	144	90	1001 0000	208	D0	1101 0000
17	11	0001 0001	DC_1	81	51	0101 0001	Q	145	91	1001 0001	209	D1	1101 0001
18	12	0001 0010	DC_2	82	52	0101 0010	R	146	92	1001 0010	210	D2	1101 0010
19	13	0001 0011	DC_3	83	53	0101 0011	S	147	93	1001 0011	211	D3	1101 0011
20	14	0001 0100	DC_4	84	54	0101 0100	T	148	94	1001 0100	212	D4	1101 0100
21	15	0001 0101	NAK	85	55	0101 0101	U	149	95	1001 0101	213	D5	1101 0101
22	16	0001 0110	SYN	86	56	0101 0110	V	150	96	1001 0110	214	D6	1101 0110
23	17	0001 0111	ETB	87	57	0101 0111	W	151	97	1001 0111	215	D7	1101 0111
24	18	0001 1000	CAN	88	58	0101 1000	X	152	98	1001 1000	216	D8	1101 1000
25	19	0001 1001	EM	89	59	0101 1001	Y	153	99	1001 1001	217	D9	1101 1001
26	1A	0001 1010	SUB	90	5A	0101 1010	Z	154	9A	1001 1010	218	DA	1101 1010
27	1B	0001 1011	ESC	91	5B	0101 1011	[	155	9B	1001 1011	219	DB	1101 1011
28	1C	0001 1100	FS	92	5C	0101 1100	\	156	9C	1001 1100	220	DC	1101 1100
29	1D	0001 1101	GS	93	5D	0101 1101	]	157	9D	1001 1101	221	DD	1101 1101
30	1E	0001 1110	RS	94	5E	0101 1110	↑	158	9E	1001 1110	222	DE	1101 1110
31	1F	0001 1111	US	95	5F	0101 1111	←	159	9F	1001 1111	223	DF	1101 1111
32	20	0010 0000	SPACE	96	60	0110 0000	`	160	A0	1010 0000	224	E0	1110 0000
33	21	0010 0001	!	97	61	0110 0001	a	161	A1	1010 0001	225	E1	1110 0001
34	22	0010 0010	"	98	62	0110 0010	b	162	A2	1010 0010	226	E2	1110 0010
35	23	0010 0011	#	99	63	0110 0011	c	163	A3	1010 0011	227	E3	1110 0011
36	24	0010 0100	$	100	64	0110 0100	d	164	A4	1010 0100	228	E4	1110 0100
37	25	0010 0101	%	101	65	0110 0101	e	165	A5	1010 0101	229	E5	1110 0101
38	26	0010 0110	&	102	66	0110 0110	f	166	A6	1010 0110	230	E6	1110 0110
39	27	0010 0111	'	103	67	0110 0111	g	167	A7	1010 0111	231	E7	1110 0111
40	28	0010 1000	(	104	68	0110 1000	h	168	A8	1010 1000	232	E8	1110 1000
41	29	0010 1001	)	105	69	0110 1001	i	169	A9	1010 1001	233	E9	1110 1001
42	2A	0010 1010	*	106	6A	0110 1010	j	170	AA	1010 1010	234	EA	1110 1010
43	2B	0010 1011	+	107	6B	0110 1011	k	171	AB	1010 1011	235	EB	1110 1011
44	2C	0010 1100	,	108	6C	0110 1100	l	172	AC	1010 1100	236	EC	1110 1100
45	2D	0010 1101	-	109	6D	0110 1101	m	173	AD	1010 1101	237	ED	1110 1101
46	2E	0010 1110	.	110	6E	0110 1110	n	174	AE	1010 1110	238	EE	1110 1110
47	2F	0010 1111	/	111	6F	0110 1111	o	175	AF	1010 1111	239	EF	1110 1111
48	30	0011 0000	0	112	70	0111 0000	p	176	B0	1011 0000	240	F0	1111 0000
49	31	0011 0001	1	113	71	0111 0001	q	177	B1	1011 0001	241	F1	1111 0001
50	32	0011 0010	2	114	72	0111 0010	r	178	B2	1011 0010	242	F2	1111 0010
51	33	0011 0011	3	115	73	0111 0011	s	179	B3	1011 0011	243	F3	1111 0011
52	34	0011 0100	4	116	74	0111 0100	t	180	B4	1011 0100	244	F4	1111 0100
53	35	0011 0101	5	117	75	0111 0101	u	181	B5	1011 0101	245	F5	1111 0101
54	36	0011 0110	6	118	76	0111 0110	v	182	B6	1011 0110	246	F6	1111 0110
55	37	0011 0111	7	119	77	0111 0111	w	183	B7	1011 0111	247	F7	1111 0111
56	38	0011 1000	8	120	78	0111 1000	x	184	B8	1011 1000	248	F8	1111 1000
57	39	0011 1001	9	121	79	0111 1001	y	185	B9	1011 1001	249	F9	1111 1001
58	3A	0011 1010	:	122	7A	0111 1010	z	186	BA	1011 1010	250	FA	1111 1010
59	3B	0011 1011	;	123	7B	0111 1011	{	187	BB	1011 1011	251	FB	1111 1011
60	3C	0011 1100	<	124	7C	0111 1100	|	188	BC	1011 1100	252	FC	1111 1100
61	3D	0011 1101	=	125	7D	0111 1101	}	189	BD	1011 1101	253	FD	1111 1101
62	3E	0011 1110	>	126	7E	0111 1110	~	190	BE	1011 1110	254	FE	1111 1110
63	3F	0011 1111	?	127	7F	0111 1111	DEL	191	BF	1011 1111	255	FF	1111 1111

Note: Boxed characters are different on national keyboards (see Appendix E).

a = Decimal
b = Hexadecimal
c = 8 bit Binary Representation
d = ASCII Code

APPENDIX E ASCII CHARACTER EQUIVALENCES

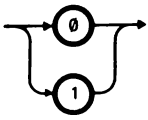
This table shows the national equivalences for those ASCII characters that appear on the video screen or printer in various national forms.

ASCII VALUE		NATIONAL EQUIVALENT												
DECIMAL	HEXADECIMAL	USA	ITALY	FRANCE	GREAT BRITAIN	GERMANY (1)	GERMANY (2)	SPAIN	PORTUGAL	DENMARK NORWAY	SWEDEN FINLAND	NORWAY	SWITZERLAND FRENCH	SWITZERLAND GERMAN
35	23	(#)	£	£	£	#	#	£	#	£	#	£	£	£
36	24	($)	$	$	$	$	$	$	$	$	¤	$	$	$
64	40	(@)	§	à	@	§	§	§	§	´	@	´	§	§
91	5B	[	°	°	[	Ä	Ä	¡	Ã	Æ	Ä	Æ	à	à
92	5C	(\)	ç	ç	\	Ö	Ö	Ñ	Ç	Ø	Ö	Ø	ç	ç
93	5D	]	é	§	]	Ü	Ü	¿	Õ	Å	Å	Å	è	è
96	60	`	ù	`	`	`	`	`	`	`		`	`	`
123	7B	{	à	é	{	ä	ä	°	ã	æ	ä	æ	ä	ä
124	7C	|	ò	ù	|	ö	ö	Ñ	ç	ø	ö	ø	ö	ö
125	7D	}	è	è	}	ü	ü	ç	º	å	å	a	ü	ü
126	7E	~	ì		~	ß	ß	~	°		~		é	é

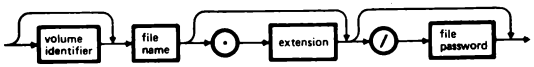
* Encircled characters are used for functions in BASIC.

APPENDIX F COMMON TERMS USED IN FORMAT SPECIFICATIONS

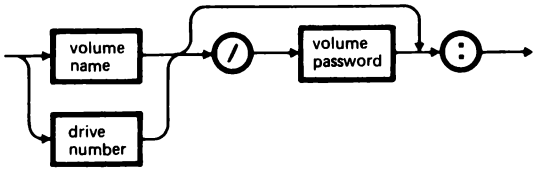
DRIVE NUMBER



FILE - IDENTIFIER



VOLUME - IDENTIFIER



Note

If drive-number is omitted, drive 0 is assumed by default unless dirve 1 was the last drive accessed.

APPENDIX G NUMERIC TYPE DECLARATION TAGS

Listed below are the tags that can be appended to a numeric variable name to declare its type.

TAG	MEANING	EXAMPLES
%	Integer	I% ROUND%
!	Single precision	A! TIME!
#	Double precision	N# SPEED#

NOTICE

Ing. C. Olivetti & C. S.p.A. reserves the right to make improvements in the product described in this manual at any time and without notice.

Anything in the standard form of the Olivetti Sales Contract to the contrary not withstanding, all software being licensed to Customer is licensed "as is". THERE ARE NO WARRANTIES EXPRESS OR IMPLIED INCLUDING WITHOUT LIMITATION THE IMPLIED WARRANTY OF FITNESS FOR PURPOSE AND OLIVETTI SHALL NOT BE LIABLE FOR ANY DIRECT, INDIRECT, CONSEQUENTIAL OR INCIDENTAL DAMAGES IN CONNECTION WITH SUCH SOFTWARE.

Printed in Italy

GR Code 3987640 R (0)

Printed in Italy