

£1.99

PHOENIX COMPUTER CRIB CARD

AMSTRAD CPC 464

KEYWORDS

OPERATING COMMANDS

GRAPHIC AND SOUND COMMANDS

COLOUR COMMANDS

BASIC STATEMENTS

BASIC FUNCTIONS

LOGICAL OPERATORS

ERROR MESSAGES

BASIC TOKENS

**EVERYTHING YOU NEED
AT YOUR FINGERTIPS**

KEY TO CONTENTS

addr	=	address
c	=	condition
ch	=	channel number
cn	=	colour number
e	=	expression
f	=	format template
i	=	integer
inc	=	increment
ln	=	line number
n	=	number
r	=	range of line numbers
s	=	stream expression
v	=	numeric variable
v\$	=	string variable or literal string
x,y	=	coordinates
[]	=	optional

PROGRAM INSTRUCTIONS

KEYWORD	SYNTAX	DEFINITION
AUTO	AUTO[ln][inc]	Automatic line numbering
CONT	CONT	Resumes program execution
DELETE	DELETE[r]	Delete lines in range
EDIT	EDITln	Amend individual program lines
LIST	LIST[r][.s]	Sends lines in range to stream
MERGE	MERGEv\$	Adds the file v\$ to program currently in memory
NEW	NEW	Completely clears the program and variables from memory
REM	REM	Remark or reminder
RENUM	RENUM [ln][.ln][.inc]	Renumbers the program
RUN	RUN[ln]	Begins executing program at line ln
TRON	TRON	Switches trace mode on
TROFF	TROFF	Switches trace mode off

GENERAL INSTRUCTIONS

KEYWORD	SYNTAX	DEFINITION
CLEAR	CLEAR	Erase all variables and arrays
DI	DI	Disables interrupts
EI	EI	Enables interrupts
END	END	Ends the program
STOP	STOP	Stops a program

STRING FUNCTIONS

KEYWORD	SYNTAX	DEFINITION
ASC	ASC(v\$)	Returns ascii value of first character in string
BINS	BINS(i[,n])	Produces string,n characters long of n in binary
CHRS	CHRS(i)	Produces ascii character i
DECS	DECS(n,f)	Generates formatted string from number
HEXS	HEXS(i[,n])	Produces string of hex digits representing i
INSTR	INSTR([i,]v\$,v\$)	Searches from a given point in a given string for another given string
LEFT\$	LEFT\$(v\$,n)	Extracts n characters from left of v\$
LEN	LEN(v\$)	Finds length of v\$
LOWERS	LOWERS(v\$)	Converts string to lower case
MIDS	MIDS(v\$,n[,n])	COMMAND:Replaces part of v\$ FUNCTION>Returns part of v\$
RIGHTS	RIGHT\$(v\$,n)	Extracts n characters from right of v\$
SPACES	SPACES(n)	Produces string of n spaces
STR\$	STR\$(n)	Value of n converted to a string
STRINGS	STRINGS(n,v\$)	Produces string consisting of n, v\$ characters
	STRINGS(n,n1)	Produces string consisting of n, ascii n1 characters
UPPER\$	UPPER\$(v\$)	Converts v\$ to upper case
VAL	VAL(v\$)	Converts string into a number

NUMBER FUNCTIONS

KEYWORD	SYNTAX	DEFINITION
ABS	ABS(n)	Returns absolute value of n
CINT	CINT(n)	Converts n to integer form
CREAL	CREAL(n)	Converts value to real form
EXP	EXP(n)	Returns the exponential of n
FIX	FIX(n)	Rounds a number towards zero
INT	INT(n)	Rounds number to nearest smaller integer
LOG	LOG(n)	Returns natural log of n
LOG10	LOG10(n)	Returns log base 10 of n
MAX	MAX(n,n,n..)	Returns largest value in list
MIN	MIN(n,n,n..)	Returns minimum value in list
RANDOMIZE	RANDOMIZE(n)	Sets the random number seed
RND	RND(n)	Returns a value in range $0 \leq n < 1$
	RND(n>0)	Returns next number in sequence
	RND(0)	Returns last random number
	RND(n<0)	Starts a new sequence based on n and returns first number
ROUND	ROUND(n[,i])	Rounds n to i decimal places
SGN	SGN(n)	Returns signum of n
SQR	SQR(n)	Returns square root of n
UNT	UNT(addr)	Converts number to integer

TRIGONOMETRIC FUNCTIONS

KEYWORD	SYNTAX	DEFINITION
ATN	ATN(n)	Returns arctangent of angle n
DEG	DEG	Sets degrees mode
PI	PI	Returns closest possible value of pi
RAD	RAD	Sets radians mode
SIN	SIN(n)	Returns sine of angle n
TAN	TAN(n)	Returns tangent of angle n

LOOP,DECISION AND CONTROL

KEYWORD	SYNTAX	DEFINITION
AFTER	AFTERi[,i] GOSUB In	Invokes subroutine after given time has elapsed
EVERY	EVERYi[,i] GOSUB In	Invokes subroutine every time a given period has elapsed
FOR	FORv =nTO n[STEPn]	Used to execute a given part of program a number of times,by stepping a control variable
GOSUB	GOSUBIn	Calls subroutine at line In
GOT	GOTOIn	Sends control to line In
IF	IFcTHENaction1 [ELSEaction2]	If c is true then action1 is executed.If c is false then action2 is executed if present otherwise the next statement is executed.
	IFcGOTOIn [ELSEaction2]	If c is true,control is passed to line In,otherwise as above
NEXT	NEXT[v[,v..]]	Delimits end of a FOR loop
ON	ONnGOSUBIn1,In2..	Chooses the nth line in list and GOSUBs that line
	ONnGOTOIn1,In2..	Chooses the nth line and GOTOS that line
	ON BREAK GOSUBIn	Enables break interrupt subroutine
	ON BREAK STOP	Turns off any trap made by ON BREAK GOSUB
	ON ERROR GOTOIn	If any error is detected,control is passed to line In
	ON ERROR GOTO0	Switches off error trapping and prints error message
	ON SQ(n)GOSUBIn	Sets sound queue interrupt
REMAIN	REMAIN(i)	Gets remaining count on timer i
RETURN	RETURN	Returns from end of subroutine
WEND	WEND	Delimits end of WHILE loop
WHILE	WHILEc	Repeats a body of program until c is false

FILE HANDLING COMMANDS

KEYWORD	SYNTAX	DEFINITION
CAT	CAT	Displays tape catalogue
CHAIN	CHAINv\$[,ln] CHAIN MERGEv\$ [[,ln][,DELETER]]	Chains v\$ and runs from line ln Same as above, but deletes lines r of current program first
CLOSEIN	CLOSEIN	Close the cassette input file
CLOSEOUT	CLOSEOUT	Close the cassette output file
EOF	EOF	Tests if at end of file
INPUT	INPUTs,list:vars	Inputs the list of variables from stream s
LINE INPUT	LINE INPUTs,v	Inputs complete line from stream s into variable
LOAD	LOADv\$[,addr]	Loads a program into memory. Goes to addr if present
MERGE	MERGEv\$	Merges v\$ from cassette with current program in memory
OPENIN	OPENINv\$	Opens cassette file v\$ for input
OPENOUT	OPENOUTv\$	Opens cassette file v\$ for output
POS	POS(s)	Returns position in stream since last carriage return
PRINT	PRINTs [,print list]	Sends print list to stream s
RUN	RUNv\$	Loads a program, v\$, from cassette and runs it
SAVE	SAVEv\$[type [,binary params]]	Saves program in memory onto cassette
SPEED WRITE	SPEED WRITEn	n=0, nominal 1000 bits per second n=1, nominal 2000 bits per second
WRITE	WRITES [,write list]	Similar to print, but data can be read back by INPUT

DATA HANDLING COMMANDS

KEYWORD	SYNTAX	DEFINITION
DATA	DATAconstants	Declares constant data
DEF	DEF INT letter range	Sets default type to Integer
	DEF REAL letter range	Sets default type to Real
	DEF STR letter range	Sets default type to String
DIM	DIMv(n,n..) DIMv\$(n,n..)	Dimensions string or numeric array, reserving memory space
ERASE	ERASElist:vars	Removes array from memory
INPUT	INPUT[s][string] list:vars	Inputs data from stream s and stores in variables
LET	[LET]v=v [LET]v\$=v\$	Defines numeric variable Defines string variable
READ	READlist:vars	Read from data statements
RESTORE	RESTORE	Restore pointer to data list

KEYBOARD AND JOYSTICK

KEYWORD	SYNTAX	DEFINITION
INKEY	INKEY(key)	Returns state of key
INKEY\$	INKEY\$	Returns next,if any,key pressed
JOY	JOY(n)	Reads status of joystick n n must be 0 or 1
KEY	KEYi,v\$	Sets new function key definition
	KEY DEFi,repeat [,norm[,shift[,ctrl]])	Defines value returned by key i
SPEED KEY	SPEED KEYstart del, repeat period	Sets the keyboard repeat speed
SYMBOL	SYMBOLn,n1,n2,..	Redefines character n with pattern n1 to n8
	SYMBOL AFTERi	Sets number of definable characters

RELATIONAL OPERATORS

SYMBOL	OPERATION
<	Less than
<= or =<	Less than or equal
=	Equal
>= or =>	Greater than or equal
>	Greater than
<>	Not equal

ARITHMETIC OPERATORS

SYMBOL	OPERATION	PRIORITY
^	Exponentation	1
-	Unary minus	2
*	Multiply	3
/	Divide	3
\	Integer division	4
MOD	Integer modulus	4
+	Addition	5
-	Subtraction	5

ISBN 0-9465-7635-1



PHOENIX PUBLISHING ASSOC. LTD
14 VERNON ROAD, BUSHEY, HERTS WD2 2JL

Copyright © Keith Hook 1985
all rights reserved

ISBN 0 9465 7635 1 (AMSTRAD CPC 464)
AMSTRAD is the reg trademark of AMSTRAD PLC



**PHOENIX
PUBLISHING
ASSOCIATES**

LOGICAL OPERATORS

KEYWORD	SYNTAX	DEFINITION
AND	iANDi cANDc	Performs bitwise operation Performs logical AND operation
OR	iORi cORc	Performs bitwise operation Performs logical OR operation
XOR	iXORi cXORc	Performs bitwise operation Performs logical EXCLUSIVE OR operation

TRUTH TABLES

FALSE	AND	FALSE	=	FALSE
FALSE	AND	TRUE	=	FALSE
TRUE	AND	FALSE	=	FALSE
TRUE	AND	TRUE	=	TRUE
FALSE	OR	FALSE	=	FALSE
FALSE	OR	TRUE	=	TRUE
TRUE	OR	FALSE	=	TRUE
TRUE	OR	TRUE	=	TRUE
FALSE	XOR	FALSE	=	FALSE
FALSE	XOR	TRUE	=	TRUE
TRUE	XOR	FALSE	=	TRUE
TRUE	XOR	TRUE	=	FALSE

GRAPHICS AND TEXT SCREENS

KEYWORD	SYNTAX	DEFINITION
BORDER	BORDERcn[,cn]	Sets the border colour and alternates between both if two are present
CLG	CLG[cn]	Clears the graphics screen with colour cn
CLS	CLS[s]	Clears screen s to paper colour
DRAW	DRAWx,y[,ink]	Draws a line on graphics screen to absolute position
DRAWR	DRAWRxo,yo[,ink]	Draws a line on graphics screen relative to current position
INK	INKink,cn[,cn]	Sets given ink to given colour
LOCATE	LOCATE[s,]x,y	Sets new cursor location
MODE	MODE[i]	Sets screen mode: 0:16 colours,25 lines,20 chars 200*160 pixels 1:4 colours,25 lines,40 chars 200*320 pixels 2:2 colours,25 lines,80 chars 200*640 pixels
MOVE	MOVEx,y MOVERxo,yo	Moves graphics cursor to x,y Moves graphics cursor relative to current position

GRAPHICS AND TEXT SCREENS

KEYWORD	SYNTAX	DEFINITION
ORIGIN	ORIGINx,y[<i>left, right,top,bottom</i>]	Set graphics screen origin and window
PAPER	PAPER[s,]ink	Set background ink for chars
PEN	PEN[s,]ink	Set foreground ink for chars
PLOT	PLOTx,y[,ink]	Plots a point on graphics screen -absolute
	PLOTRxo,yo[,ink]	Plots a point on graphics screen -relative
SPEED INK	SPEED INKperiod,	Sets time for flashing inks period
TAG	TAG[s]	Writes text to graphics cursor
TAGOFF	TAGOFF[s]	Cancels TAG
TEST	TEST(x,y)	Test ink at given graphics position
	TESTR(xo,yo)	Test ink at point relative to current graphics position
VPOS	VPOS(s)	Returns position of cursor from top of window
WINDOW	WINDOW[s,] <i>left right,top,bottom</i>	Sets text window screen
	WINDOW SWAPs,s2	Swaps text windows
XPOS	XPOS	Gets x graphics position
YPOS	YPOS	Gets y graphics position

SOUND CHANNELS

KEYWORD	SYNTAX	DEFINITION
ENT	ENTenv[<i>list: envelope sections</i>]	Sets up tone envelope
ENV	ENVenv[<i>list: envelope sections</i>]	Sets up volume envelope
RELEASE	RELEASEch	Releases sound channels from hold
SOUND	SOUNDchan stat,tone [,dur[,vol[,vol env [,tone env[,noise period]]]]]	Puts a sound command into sound queue

PRINT RELATED COMMANDS

KEYWORD	SYNTAX	DEFINITION
PRINT	PRINT[s][<i>print list</i>][<i>using f</i>] [<i>separator</i>]	Prints the list on stream s
WIDTH	WIDTHi	Sets the width of the printer in characters
ZONE	ZONEi	Sets the print zone size

MACHINE LEVEL OPERATORS

KEYWORD	SYNTAX	DEFINITION
CALL	CALLn[,list: params]	Calls external subroutine
HIMEM	HIMEM	Address of highest byte used by BASIC
INP	INP(addr)	Reads value from given I/O port
MEMORY	MEMORY(addr)	Sets highest byte usable by BASIC
OUT	OUTaddr,i	Sends value to given I/O port
PEEK	PEEK(addr)	Looks at memory location
POKE	POKEaddr,i	Puts a value into memory
WAIT	WAITaddr,i[,i2]	Waits until given I/O port returns a particular value

CONTROL CHARACTERS

VALUE	NAME	MEANING
&02	2	STX Turn off text cursor
&03	3	ETX Turn on text cursor
&04	4	EOT 0..2 Set screen mode
&06	6	ACK Enable text screen
&07	7	BEL Sound bleeper
&08	8 *	BS Move cursor back one character
&09	9 *	TAB Move cursor forward one character
&0A	10 *	LF Move cursor down one line
&0B	11 *	VT Move cursor up one line
&0C	12	FF Clear text window.Same as CLS
&0D	13 *	CR Move cursor to left edge of line
&0E	14	SO 0..15 Set paper ink
&0F	15	SI 0..15 Set pen ink
&10	16 *	DLE Delete current character
&15	21	NAK Turn off text screen
&16	22	SYN 0..1 Enables/disables transparent option
&18	24	CAN Exchange paper and pen inks

ERROR HANDLING

KEYWORD	SYNTAX	DEFINITION
ERR	ERR	Error number of last error
ERL	ERL	Error line of last error
ERROR	ERRORi	Causes error action to be taken.
RESUME		Only allowed in error proc.mode Returns control to start of statement in which error occurred
	RESUMEIn	Returns control to line In
	RESUME NEXT	Returns control to statement after error line

ERROR MESSAGES

- 1 – Unexpected NEXT
- 2 – Syntax error
- 3 – Unexpected RETURN
- 4 – DATA exhausted
- 5 – Improper argument
- 6 – Overflow
- 7 – Memory full
- 8 – Line doesn't exist
- 9 – Subscript out of range
- 10 – Array already dimensioned
- 11 – Division by zero
- 12 – Invalid direct command
- 13 – Type mismatch
- 14 – String space full
- 15 – String too long
- 16 – String expression too complex
- 17 – Cannot CONTINUE
- 18 – Unknown user function
- 19 – RESUME missing
- 20 – Unexpected RESUME
- 21 – Direct command found
- 22 – Operand missing
- 23 – Line too long
- 24 – EOF met
- 25 – File type error
- 26 – NEXT missing
- 27 – File already open
- 28 – Unknown command
- 29 – WORD missing
- 30 – Unexpected WORD

COLOURS

- | | |
|--------------------|---------------------|
| 0 – Black | 14 – Pastel blue |
| 1 – Blue | 15 – Orange |
| 2 – Bright blue | 16 – Pink |
| 3 – Red | 17 – Pastel magenta |
| 4 – Magenta | 18 – Bright green |
| 5 – Mauve | 19 – Sea green |
| 6 – Bright red | 20 – Bright cyan |
| 7 – Purple | 21 – Lime |
| 8 – Bright magenta | 22 – Pastel green |
| 9 – Green | 23 – Pastel cyan |
| 10 – Cyan | 24 – Bright yellow |
| 11 – Skyblue | 25 – Pastel yellow |
| 12 – Yellow | 26 – Bright white |
| 13 – White | |

BASIC TOKENS

KEYWORD	TOKEN	KEYWORD	TOKEN
ABS	FF00	AFTER	80
ASC	FF01	ATN	FF02
AUTO	81	BIN\$	FF71
BORDER	82	CALL	83
CAT	84	CHAIN	85
CHAIN MERGE	85 AB	CHR\$	FF03
CINT	FF04	CLEAR	86
CLG	87	CLOSE IN	88
CLOSEOUT	89	CLS	8A
CONT	8B	COS	FF05
CREAL	FF06	DATA	8C
DEF FN	8D	DEFINT	8E
DEFREAL	8F	DEFSTR	90
DEG	91	DELETE	92
DIM	93	DRAW	94
DRAWR	95	EDIT	96
ELSE	97	END	98
ENT	99	ENV	9A
EOF	FF4D	ERASE	9B
ERR	FF41	ERL	E3
ERROR	9C	EVERY	9D
EXP	FF07	FIX	FF08
FOR	9E	FRE	FF09
GOSUB	9F	GOTO	A0
HEX\$	FF73	HIMEM	FF42
IF	A1	INK	A2
INKEY	FF0A	INKEY\$	FF43
INP	FF0B	INPUT	A3
INSTR	FF74	INT	FF0C
JOY	FF0D	KEY	A4
KEY DEF	FF75	LEFT\$	DB
LEN	FF0E	LET	A5
LINE INPUT	A6 A3	LIST	A7
LOCATE	A9	LOG	FF0F
LOG10	FF10	LOWER\$	FF11
MAX	FF76	MEMORY	AA
MERGE	AB	MID\$	AC
MIN	FF77	MODE	AD
MOVE	AE	MOVER	AF
NEW	B1	NEXT	B0

BASIC TOKENS

KEYWORD	TOKEN	KEYWORD	TOKEN
ON GOSUB	B220	ON GOTO	B2A0
ON BREAK GOSUB	B39F	ON BREAK STOP	B3CE
ON ERROR GOTO	B29C	ON SQ GOSUB	B59F
OPENIN	B6	OPENOUT	B7
ORIGIN	B8	OUT	B9
PAPER	BA	PEEK	FF12
PEN	BB	PI	FF44
PLOT	BC	PLOTR	BD
POKE	BE	POS	FF78
PRINT	BF	RAD	C1
RANDOMIZE	C2	READ	C3
RELEASE	C4	REM	C5
RENUM	C6	RESTORE	C7
RESUME	C8	RETURN	C9
RIGHT\$	FF79	RND	FF45
ROUND	FF7A	RUN	CA
SAVE	CB	SGN	FF14
SIN	FF15	SOUND	CC
SPACE\$	FF16	SPEED INK	CD A2
SPEED KEY	CD A4	SPEED WRITE	CD D9
SQ	FF17	SQR	FF18
STOP	CE	STR\$	FF19
STRING\$	FF73	SYMBOL	CF
SYMBOL AFTER	CF 80	TAG	D0
TAGOFF	D1	TAN	FF1A
TEST	FF7C	TESTSR	FF7D
TIME	FF46	TROFF	D2
TRON	D3	UNT	FF1B
UPPER\$	FF1C	VAL	FF1D
VPOS	FF7F	WAIT	D4
WEND	D5	WHILE	D6
WIDTH	D7	WINDOW	D8
WINDOW SWOP	D8 E7	WRITE	D9
XPOS	FF47	YPOS	FF48
ZONE	DA	=	EF
<	F2	+	F4
-	F5	*	F6
/	F7		

£1.99

PHOENIX COMPUTER CRIB CARD

AMSTRAD CPC 464

KEYWORDS

OPERATING COMMANDS

GRAPHIC AND SOUND COMMANDS

COLOUR COMMANDS

BASIC STATEMENTS

BASIC FUNCTIONS

LOGICAL OPERATORS

ERROR MESSAGES

BASIC TOKENS

EVERYTHING YOU NEED
AT YOUR FINGERTIPS

120

6

KEY TO CONTENTS

addr	...	address
c	...	condition
ch	...	channel number
cn	...	colour number
e	...	expression
f	...	format
inc	...	increment
ln	...	line number
n	...	number
r	...	range of line numbers
s	...	stream expression
v	...	numeric variable
x,y	...	string variable or literal string
[]	...	coordinates
	...	optional

PROGRAM INSTRUCTIONS

KEYWORD	SYNTAX	DEFINITION
AUTO	AUTO[<i>n</i>][<i>inc</i>]	Automatic line numbering
CONT	CONT	Resumes program execution
DELETE	DELETE[<i>v</i>]	Deletes lines in range
EDIT	EDIT[<i>n</i>]	Amend individual program lines
LIST	LIST[<i>v</i>][<i>l</i>]	Sends lines in range to stream
MERGE	MERGE <i>v</i>	Adds the file <i>v</i> to program currently in memory
NEW	NEW	Compactly clears the program and variables from memory
REM	REM	Remark or noncode
RENUM	RENUM[<i>n</i>][<i>inc</i>][<i>inc</i>]	Renumbers the program
RUN	RUN[<i>n</i>]	Begins executing program at line <i>n</i>
TRON	TRON	Switches trace mode on
TROFF	TROFF	Switches trace mode off

GENERAL INSTRUCTIONS

KEYWORD	SYNTAX	DEFINITION
CLEAR	CLEAR	Erase all variables and arrays
DI	DI	Disables interrupts
EI	EI	Enables interrupts
END	END	Ends the program
STOP	STOP	Stops a program

STRING FUNCTIONS

KEYWORD	SYNTAX	DEFINITION
ASC	ASC(<i>v</i>)	Returns ASCII value of first character in string
ASCN	ASCN[<i>n</i>](<i>v</i>)	Produces string of characters long of <i>n</i> in string <i>v</i>
BINS	BINS(<i>i</i>)	Produces ASCII character <i>i</i>
CHAS	CHAS(<i>n</i>)	Generates formatted string from number
DECS	DECS(<i>n</i>)	Produces string of hex digits representing <i>n</i>
INSTR	INSTR[<i>n</i>](<i>v</i> 1, <i>v</i> 2)	Searches from a given point in a given string for another given string
LEFTS	LEFTS(<i>v</i> 1)	Extracts <i>n</i> characters from left of <i>v</i>
LEN	LEN(<i>v</i>)	Finds length of <i>v</i>
LOWERS	LOWERS(<i>v</i>)	Converts string to lower case
MID	MID(<i>v</i> 1, <i>v</i> 2, <i>n</i>)	COMMAND-Extracts part of <i>v</i>
RIGHTS	RIGHTS(<i>v</i> 1)	Extracts <i>n</i> characters from right of <i>v</i>
SPACES	SPACES(<i>n</i>)	Produces string of <i>n</i> spaces
STRN	STRN(<i>v</i>)	Value of <i>v</i> converted to a string
STRINGS	STRINGS(<i>v</i> 1, <i>v</i> 2)	Produces string consisting of <i>n</i> , <i>v</i> characters
STRINGS(L)	STRINGS(L, <i>n</i> 1)	Produces string containing <i>n</i> , <i>v</i> and <i>l</i> characters
UPPERS	UPPERS(<i>v</i>)	Converts <i>v</i> to upper case
VAL	VAL(<i>v</i>)	Converts string into a number

NUMBER FUNCTIONS

KEYWORD	SYNTAX	DEFINITION
ABS	ABS(<i>n</i>)	Returns absolute value of <i>n</i>
ABS	ABS(<i>n</i>)	Converts <i>n</i> to integer form
CINT	CINT(<i>n</i>)	Converts value to real form
CREAL	CREAL(<i>n</i>)	Returns the exponential of <i>n</i>
EXP	EXP(<i>n</i>)	Rounds a number towards zero
FIX	FIX(<i>n</i>)	Rounds number to nearest smaller integer
INT	INT(<i>n</i>)	
LOG	LOG(<i>n</i>)	Returns natural log of <i>n</i>
LOG10	LOG10(<i>n</i>)	Returns base 10 of <i>n</i>
MAX	MAX(<i>n</i> 1, <i>n</i> 2, <i>n</i> 3)	Returns largest value in list
MIN	MIN(<i>n</i> 1, <i>n</i> 2, <i>n</i> 3)	Returns minimum value in list
RANDOMIZE	RANDOMIZE(<i>n</i>)	Sets the random number seed
RND	RND(<i>n</i>)	Returns a value in range 0 - <i>n</i> - 1
RND	RND(<i>n</i> 1, <i>n</i> 2)	Returns random number in sequence
RND	RND(<i>n</i>)	Returns last random number
RND	RND(<i>n</i> 1, <i>n</i> 2)	Starts a new sequence based on <i>n</i> and returns first number
ROUND	ROUND(<i>v</i> 1, <i>i</i>)	Rounds <i>n</i> to <i>i</i> decimal places
SGN	SGN(<i>n</i>)	Returns signum of <i>n</i>
SGR	SGR(<i>n</i>)	Returns square root of <i>n</i>
UNT	UNT(<i>i</i>)	Converts number to integer

TRIGONOMETRIC FUNCTIONS

KEYWORD	SYNTAX	DEFINITION
ATN	ATN(<i>n</i>)	Returns arctan of <i>n</i>
DEG	DEG(<i>n</i>)	Returns degree mode
PI	PI	Returns constant pi
RAD	RAD(<i>n</i>)	Returns radian mode
SIN	SIN(<i>n</i>)	Returns sine of angle <i>n</i>
TAN	TAN(<i>n</i>)	Returns tangent of angle <i>n</i>

LOOP, DECISION AND CONTROL

KEYWORD	SYNTAX	DEFINITION
AFTER	AFTER[<i>i</i>]	Invokes subroutine after given time has elapsed
EVERY	EVERY[<i>i</i>]	Invokes subroutine every <i>i</i> lines
FOR	FOR <i>n</i> IN <i>GOSUB</i> <i>n</i> 1 TO <i>n</i> 2 STEP <i>n</i> 3	Used to execute a given part of program a number of times by stepping a control variable
GOSUB	GOSUB <i>n</i>	Calls subroutine at line <i>n</i>
GOT	GOT <i>n</i>	Sends control to line <i>n</i>
IF	IF (<i>EL</i> <i>EL</i> <i>EL</i>)	If <i>n</i> is true then action is executed. If <i>n</i> is false then action is executed if present otherwise the next statement is executed.
IFGOT	IFGOT <i>n</i>	If <i>n</i> is true control is passed to line <i>n</i> otherwise no action
ELSE	ELSE <i>EL</i> <i>EL</i> <i>EL</i>	Delimits end of a FOR loop
NEXT	NEXT <i>n</i>	Delimits end of a FOR loop and GOSUBS that line
ON	ON <i>GOSUB</i> <i>n</i> 1 TO <i>n</i> 2	Chooses the <i>n</i> th line in list
ON	ON <i>GOT</i> <i>n</i> 1 TO <i>n</i> 2	Chooses the <i>n</i> th line and GOTs that line
ON BREAK	ON BREAK GOSUB <i>n</i>	Enables break interrupt
ON BREAK	ON BREAK STOP	Turns off any trap made by ON BREAK GOSUB
ON ERROR	ON ERROR GOT <i>n</i>	If any error is detected control is passed to line <i>n</i>
ON ERROR	ON ERROR GOT <i>n</i>	Switches off error trapping and prints error message
ON SQ	ON SQ IN <i>GOSUB</i> <i>n</i>	Sets sound queue interrupt
REMAIN	REMAIN(<i>n</i>)	Gets remaining count on timer
RETURN	RETURN	Returns from end of subroutine
WHILE	WHILE <i>EL</i> <i>EL</i> <i>EL</i>	Delimits end of WHILE loop
WHILE	WHILE <i>EL</i> <i>EL</i> <i>EL</i>	Repeats a body of program until <i>n</i> is false

FILE HANDLING COMMANDS

KEYWORD	SYNTAX	DEFINITION
CAT	CAT <i>FILE</i>	Displays tape catalogue
CHAIN	CHAIN <i>FILE</i>	Opens <i>v</i> and runs from line <i>n</i> of current program file
CHDIR	CHDIR <i>FILE</i>	Change the current input file
CLOSE	CLOSE <i>FILE</i>	Closes the file of <i>FILE</i>
CLOSIO	CLOSIO <i>FILE</i>	Tests the end of file
EOF	EOF	Inputs the last of variables from stream <i>n</i>
EXEC	EXEC <i>FILE</i>	Inputs complete line from stream <i>n</i> into variable
LINE INPUT	LINE INPUT <i>FILE</i>	Loads a program into memory
LOAD	LOAD <i>FILE</i>	Loads a program into memory
MERGE	MERGE <i>FILE</i>	Merges <i>v</i> from cassette with <i>n</i> into <i>FILE</i>
OPEN	OPEN <i>FILE</i>	Opens cassette file <i>v</i> for output
OPENIO	OPENIO <i>FILE</i>	Returns position in stream since last carriage return
PRINT	PRINT <i>FILE</i>	Sends print list to stream <i>n</i>
RUN	RUN <i>FILE</i>	Loads a program <i>v</i> from cassette and runs it
SAVE	SAVE <i>FILE</i>	Saves program in memory onto cassette
SPACE	SPACE <i>FILE</i>	<i>n</i> - Character 1000 bits per second
WRITE	WRITE <i>FILE</i>	Similar to print but data can be read back by INPUT

DATA HANDLING COMMANDS

KEYWORD	SYNTAX	DEFINITION
DATA	DATA <i>CONST</i>	Declares constant data
DEF	DEF <i>FILE</i>	Sets default type to integer
DEF REAL	DEF <i>FILE</i>	Set default type to Real
DEF STR	DEF <i>FILE</i>	Set default type to String
DN	DN <i>FILE</i>	Dimensions string or numeric array
ERASE	ERASE <i>FILE</i>	Removes array from memory
INPUT	INPUT <i>FILE</i>	Inputs data from stream <i>n</i> and stores in variable
LET	LET <i>FILE</i> = <i>v</i>	Defines numeric variable
READ	READ <i>FILE</i>	Reads string variable
RESTORE	RESTORE	Returns pointer to data list