

PARAMETERS

a	=	a numeric expression or value to be converted into a string.	
a\$	=	a string holding numeric value or expression to be converted to a numeric variable or number.	
addr	=	address	(0-65536)
ang	=	angle (in radians)	
b	=	binary number	
c	=	condition	
cn	=	colour number	(0 to 7)
col	=	column	(0 to 31)
d	=	duration	(1=1 second)
e	=	expression	
f	=	device number	f = 0 keyboard f = 1 bottom two lines of screen f = 2 screen lines 0 to 21 f = 3 printer
i	=	integer	
inc	=	increment	
item	=	print items	
len	=	length	
l\$	=	string literal of one graphics character e.g. "P"	
lin	=	line	(0 to 21)
ln	=	line number	
n,j,k	=	number	
p	=	pitch	(0 to 69)
rn	=	row number	(0 to 7)
r	=	row	
rad	=	radius	
sadd	=	start address	(0 to 65536)
st	=	stream number	(0 to 15)
v	=	numeric variable	
v\$	=	string variable/literal string	

£1.99

PHOENIX COMPUTER CRIB CARD

SPECTRUM

KEYWORDS

OPERATING COMMANDS

GRAPHIC AND SOUND COMMANDS

COLOUR COMMANDS

DATA COMMANDS

INPUT/OUTPUT COMMANDS

MICRODRIVE COMMANDS

THE KEYBOARD

KEYWORD	CURSORS	KEY
ABS	E	G
ACS	E	SHIFT W
AND	K,L,C	SHIFT Y
ASN	E	SHIFT Q
AT	K,L,C	SHIFT I
ATN	E	SHIFT E
ATTR	E	SHIFT L
BEEP	E	SHIFT Z
BIN	E	B
BORDER	K	B
BRIGHT	E	SHIFT B
CAPS LOCK	K,L	C/SH 2
CAT	E	SHIFT 9
CHR\$	E	U
CIRCLE	E	SHIFT H
CLEAR	K	X
CLOSE #	K	SHIFT 5
CLS	K	V
CODE	K	I
CONTINUE	K	C
COPY	K	Z
COS	E	W
DATA	E	D
DEF FN	E	SHIFT 1
DELETE	C,G	0
DELETE	K,L,C,G	C/SH 0
DIM	K	D
DRAW	K	W
EDIT	K,L,C	C/SH 1
ERASE	E	SHIFT 7
EXP	E	X
FLASH	E	SHIFT V
FN	E	SHIFT 2
FOR	K	F
FORMAT	E	SHIFT 0
GO SUB	K	H
GO TO	K	G
GRAPHIC CURSOR	K,L,C	C/SH 9
IF	K	U
IN	E	SHIFT I
INK	E	SHIFT X
INKEY\$	E	N
INPUT	K	I
INT	E	R
INVERSE	E	SHIFT M
LEN	E	K
LET	K	L
LINE	E	SHIFT 3
LIST	K	K
LLIST	E	V
LN	E	Z
LOAD	K	J
LPRINT	E	C
MERGE	E	SHIFT T
MOVE	E	SHIFT 6
NEW	K	A
NEXT	K	N
NOT	K,L,C	SHIFT S
OPEN #	E	SHIFT 4
OR	K,L,C	SHIFT U
OUT	E	SHIFT O
OVER	E	SHIFT N
PAPER	E	SHIFT C
PAUSE	K	M
PEEK	E	O

KEYWORD	SYNTAX	DEFINITION
OPERATING COMMANDS		
CLEAR	CLEAR [n]	Clears variables [changes position of RAMTOP]
CONTINUE	CONTINUE	Continues execution of program after a BREAK or STOP command
EDIT	EDIT	Edits line indicated by the > cursor
LOAD	LOAD ""	Loads first program encountered on the tape into memory
	LOAD v\$	Loads program called v\$ into memory
	LOAD v\$ CODE [[sadd],[len]]	Loads v\$. Loads len bytes into memory starting at address sadd
	LOAD v\$ DATA v[\$]()	Loads string or numeric array into memory
	LOAD v\$ SCREEN\$	Loads screen display v\$ into memory
MERGE	MERGE v\$	Loads and merges v\$ with the program currently residing in memory
NEW	NEW	Clears program and variables
RUN	RUN [ln]	Executes program [from ln]
SAVE	SAVE v\$	Saves the program called v\$ on tape
	SAVE v\$ CODE sadd,len	Saves v\$ on tape. Saves len bytes starting at address sadd
	SAVE v\$ DATA v[\$]()	Saves string or numeric array
	SAVE v\$ SCREEN\$	Saves screen display v\$ on tape
STOP	STOP	Stops execution of a program
VERIFY	VERIFY v\$	Verifies that saving v\$ on tape was successful
	VERIFY v\$ CODE sadd,len	Verifies v\$. Verifies that len bytes have been saved
	VERIFY v\$ DATA	Verifies that the array

KEYWORD	SYNTAX	DEFINITION
NUMBER FUNCTIONS		
ABS	ABS n	Returns absolute value of n
BIN	BIN b	Returns decimal value of b. b cannot be a variable
DEF FN	DEF FNv(v,v,...)	Defines user-defined function
EXP	EXP n	Returns $e \wedge n$
FN	FNv(v,v,...)	Calls user-defined function
INT	INT n	Truncates n to return an integer value
LN	LN n	Returns the natural logarithm of n
PI	PI	3.1415927
RANDOMIZE	RANDOMIZE [n]	Random number seed value n=0 to 65535
RND	RND	Returns a pseudo-random number
SGN	SGN n	Returns signum of n
SQR	SQR n	Returns square root of n

TRIGONOMETRIC FUNCTIONS

ACS	ACS n	Returns arc-cosine of n
ASN	ASN n	Returns arc-sine of n
ATN	ATN n	Returns arc-tangent of n
COS	COS n	Returns cosine of n
SIN	SIN n	Returns sine of n
TAN	TAN n	Returns tangent of n

ARITHMETIC OPERATORS

symbol	operation	priority
+	addition	4

KEYWORD	SYNTAX	DEFINITION
LOGICAL OPERATORS		
AND	c AND c	Combines relations only returning True (1) if both conditions are True. j AND k=j if k<>0 or 0 if k=0 a\$ AND k=a\$ if k<>0 or "" if k=0
IF	IF c THEN action	A decision structure. The condition must be met (True) before the THEN statement is executed.
NOT	NOT e	Returns the logical inverse of an expression. NOT e=0 if e<>0 or 1 if e=0
OR	c OR c	Combines relations returning True (1) if either of its conditions are True.
THEN	IF c THEN action	The action is only executed if the condition is met

LOOPS		
FOR	FOR v=n TO n1	A loop structure. Calls subroutine starting at line ln, storing return address Transfers control of program to line ln Terminates loop when v=n1 (end value of the loop counter) Returns control to the line following the GOSUB statement Determines the increment value. If omitted inc=1
GOSUB	GOSUB ln	
GOTO	GOTO ln	
NEXT	NEXT v	
RETURN	RETURN	
STEP	FOR v=n TO n1 [STEP inc]	

KEYWORD**SYNTAX****DEFINITION****I/O INSTRUCTIONS**

AT	PRINT AT lin,col[;item]	Moves print position to the point specified by lin,col
CLS	CLS	Clears the screen
COPY	COPY	Lists a COPY of the screen on the printer
IN	IN addr	Returns the byte read from the I/O port (addr).
INKEY\$	V\$=INKEY\$	Returns key-press. Does not wait for input
INPUT	INPUT v,v\$,...	INPUT numeric or string data
	INPUT LINE v\$	Allows a string to be INPUT without quotes
INPUT #	INPUT # f;v,v\$,...	Inputs string or numeric data
INPUT #	INPUT # f; LINE v\$	Allows the input of string data without quotes to device specified by f
LINE	INPUT LINE v\$	Allows a string to be INPUT without quotes
	SAVE v\$ LINE ln	SAVEs a program storing the line number from which it runs automatically when loaded
LIST	LIST [ln]	Lists program to the screen
LIST #	LIST # f[,ln]	Lists program to device specified by f
LLIST	LLIST [ln]	Lists program on the printer
LPRINT	LPRINT v\$,v,...	Prints argument on the printer
OUT	OUT addr,n	Writes value n to I/O port addr
PRINT	PRINT [item] PRINT AT lin,col[; item] PRINT TAB col[; item]	Prints to the screen Prints item at the point specified by lin,col Prints item at the point specified by col
PRINT #	PRINT # f[[;item], item]	Prints to device f

KEYWORD	SYNTAX	DEFINITION
MICRODRIVE COMMANDS		
FORMAT	FORMAT "m";n;v\$	Formats cartridge in drive n and names it v\$
	FORMAT "n";n	Sets network station number to n
	FORMAT "t";n	Sets baud rate for RS232 interface to n
	FORMAT "b";n	Sets baud rate for RS232 interface to n
INKEY\$#	INKEY\$# st	Returns a single character from stream st, of "" if no character is available
MOVE	MOVE source TO des	Moves data from source to des (destination)
OPEN#	OPEN# st,n	Enables I/O to channel n of stream st

GRAPHICS AND COLOUR COMMANDS

ATTR	ATTR (lin,col)	Returns the colour attribute at the point lin,col.
BORDER	BORDER cn	Makes border the colour cn
BRIGHT	BRIGHT z	Brightness control, z=0 normal, z=1 bright, z=8 transparent
CIRCLE	CIRCLE x,y,rad	Draws a circle with centre x,y and radius rad
DRAW	DRAW x,y[,ang]	Draws line from previous point to relative x,y co-ordinates [turning through angle ang]
FLASH	FLASH z	Controls flashing, z=0 normal, z=1 flash, z=8 no change
INK	INK cn	Makes the ink the colour cn, cn=8 no change

KEYWORD	SYNTAX	DEFINITION
DATA COMMANDS		
DATA	DATA v,v,....	Stores data items: numeric
	DATA v\$,v\$,...	Stores data items: string
	DATA v,v\$,...	Stores data items: both numeric and string
DIM	DIM v(n[,n..])	Dimensions numeric array
	DIM v\$(n[,n,..])	Dimensions string array
LET	LET v=n	Defines numeric variable
	LET v\$=v\$	Defines string variable
READ	READ v[,v,.]	Reads numeric data into v
	READ v\$[,v\$,...]	Reads string data into v\$
	READ v[\$](n)	Reads string/numeric data into the array
RESTORE	RESTORE [ln]	Restores data. Moves pointer to the start of data [or line ln]

USEFUL POKES & PEEKS

POKE 23562,3	Key repeat
POKE 23561,n	n/50 of a second delay before key repeat
POKE 23609,n	louder click, n=1 to 15
POKE 23609,n	louder beep, n=15 to 255
POKE 23609,0	disables to normal click
POKE 23692,0	auto scroll
POKE 23659,n	stops printing on bottom n lines, n=2 to 25
POKE 23659,1	print on 23 lines
POKE 23659,0	print on 24 lines
POKE (PEEK 23635+256*PEEK 23636),n	Protects a program, n=40 to 63 will change previous lines to a symbol

ATTRIBUTE FILE

Each character display location is complemented by an attribute location which determines display mode (i.e. INK, PAPER, BRIGHT etc). The start address for the attribute is 23528 and the end address 23295. By POKEing the appropriate value into the required location the display features of a given character can be set. The codes below set paper and ink. For example:

POKE 23295,2

Will set INK to red and PAPER to black. For BRIGHT simply add 64 to any code:

POKE 23295,68

and to create the FLASH add a further 128:

POKE 23295,196

The intersection point in the table below establishes the initial INK/PAPER code:

INK COLOUR	PAPER COLOUR							
	BLACK	BLUE	RED	MAGENTA	GREEN	CYAN	YELLOW	WHITE
BLACK	0	8	16	24	32	40	48	56
BLUE	1	9	17	25	33	41	49	57
RED	2	10	18	26	34	42	50	58
MAGENTA	3	11	19	27	35	43	51	59
GREEN	4	12	20	28	36	44	52	60
CYAN	5	13	21	29	37	45	53	61
YELLOW	6	14	22	30	38	46	54	62
WHITE	7	15	23	31	39	47	55	63

DISPLAY FILE

The full Spectrum screen makes use of 6144 locations for character display and, since each character requires eight locations, individual symbols can be displayed in a total of 768 print positions (6144/8). In order to position a character on the 24 x 32 screen we require the start address for the position in question. The table which follows provides the start address for the first and last character position of each of the twenty-four rows:

ERROR MESSAGES

These appear at the bottom of the screen whenever the computer stops executing some BASIC, and explain why it stopped.

The Error Messages have a code number or letter for you to refer to.

CODE	MEANING	PROBABLE REASONS
1	NEXT without FOR The control variable does not exist (it has not been set up by a FOR statement), but there is an ordinary variable with the same name.	NEXT
2	Variable not found For a simple variable this will happen if the variable is used before it has been defined by a LET, READ or INPUT statement loaded from tape or set up in a FOR statement.	Any
3	Subscript wrong A subscript is beyond the dimension of the array, or there are the wrong number of subscripts.	Subscripted variables, Substrings
4	Out of memory There is not enough room in the computer for what you are trying to do.	LET, INPUT, FOR, DIM, GO SUB, LOAD, MERGE. Sometimes during expression evaluation
5	Out of screen An INPUT statement has tried to generate more than 23 lines in the lower half of the screen. Also occurs with PRINT AT 22,...	INPUT, PRINT AT
6	Number too big	Any arithmetic
7	RETURN without GO SUB There has been one more RETURN than there were GO SUBs.	RETURN
9	STOP statement	STOP

ERROR MESSAGES

These appear at the bottom of the screen whenever the computer stops executing some BASIC, and explain why it stopped.

The Error Messages have a code number or letter for you to refer to.

CODE	MEANING	PROBABLE REASONS
E	Out of DATA You have tried to READ past the end of the DATA list.	READ
F	Invalid file name SAVE with name empty or longer than 10 characters	SAVE
G	No room for line There is not enough room left in memory to accommodate the new program line.	Entering a line into the program
H	STOP in INPUT Some INPUT data started with STOP, or INPUT LINE was pressed.	INPUT
I	FOR without NEXT There was a FOR loop to be executed and the corresponding NEXT statement could not be found.	FOR
J	Invalid I/O device	Microdrive, etc, operations
K	Invalid colour The number specified is not an appropriate value.	INK, PAPER, BORDER, FLASH, BRIGHT, INVERSE, OVER; also after one of the corresponding control characters
L	BREAK into program BREAK pressed, this is detected between two statements.	Any
M	RAMTOP no good The number specified for RAMTOP is either too big or too small.	CLEAR; possibly in RUN

£1.99

PHOENIX COMPUTER CRIB CARD

SPECTRUM

KEYWORDS

OPERATING COMMANDS

GRAPHIC AND SOUND COMMANDS

COLOUR COMMANDS

DATA COMMANDS

INPUT/OUTPUT COMMANDS

MICRODRIVE COMMANDS