**Operators**

Each operator or group of operators is precedent over the group below it.

<table>
<thead>
<tr>
<th>Operator</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>↑</td>
<td>Exponentiation</td>
</tr>
<tr>
<td>−, +</td>
<td>Unary negative, positive</td>
</tr>
<tr>
<td>*, /</td>
<td>Multiplication, division</td>
</tr>
<tr>
<td>+, −, =, =, =, =, &gt;, &lt;</td>
<td>Addition and concatenation, subtraction</td>
</tr>
<tr>
<td>NOT</td>
<td>Relational tests</td>
</tr>
<tr>
<td>AND</td>
<td></td>
</tr>
<tr>
<td>OR</td>
<td></td>
</tr>
</tbody>
</table>

**Graphic Character Codes**

Given the color (1-8) and the pattern (0-15), this formula will generate the correct code:

\[
\text{code} = 128 + 16 \times (\text{color} - 1) + \text{pattern}
\]

<table>
<thead>
<tr>
<th>Color</th>
<th>Pattern</th>
<th>Code</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>0</td>
<td>1</td>
<td>1</td>
</tr>
<tr>
<td>0</td>
<td>2</td>
<td>2</td>
</tr>
<tr>
<td>...</td>
<td>...</td>
<td>...</td>
</tr>
<tr>
<td>12</td>
<td>13</td>
<td>251</td>
</tr>
<tr>
<td>12</td>
<td>14</td>
<td>252</td>
</tr>
<tr>
<td>12</td>
<td>15</td>
<td>253</td>
</tr>
</tbody>
</table>

For example, to print pattern 9 in blue (code 3), type:

\[
C = 128 + 16 \times (3 - 1) + 9
\]

\[
? \text{CHR\$}(C)
\]

**Functions**

- **ARG** (numeric) Computes the argument of a complex number.
- **ASC** (str) Returns ASCII code of first character of specified string.
- **ATN** (numeric) Returns arctangent in radians.
- **CHRS** (code) Returns character for ASCII, control, or graphics code.
- **COS** (numeric) Returns cosine of an angle given in radians.
- **EOF** (f) Returns FALSE (0) if there is more data; TRUE(-1) if end of file has been read. For cassette, f = -1; for keyboard, f = 0.
- **EXP** (numeric) Returns natural exponential of number.
- **FIX** (numeric) Returns truncated (whole number) value.
- **HEXS** (numeric) Computes hexadecimal value.
- **INKEYS** Checks the keyboard and returns the key being pressed (if any).
- **INT** (numeric) Converts a number to an integer.
JOYSTK (j)  Returns the horizontal or vertical coordinate (j) of
the left or right joystick.
0 = horizontal, left joystick
1 = vertical, left joystick
2 = horizontal, right joystick
3 = vertical, right joystick
M = JOYSTK (K)
H = JOYSTK (K)

LEFTS (str.length)  Returns left portion (length characters) of
string.
$ = LEFTS (M$ : 7)

LEN (str)  Returns the length of a string.
X = LEN (GEN$)

LOG (numeric)  Returns natural logarithm.
Y = LOG (353)

MEM  Finds the amount of free memory.
PRINT MEM

MIDS (str, pos, length)  Returns a substring of another string
starting at pos. If length is omitted, the entire string right
of position is returned.
F$ = MIDS (A$ : 3)
?MIDS (A$ : 3, 2)

PEEK (location)  Returns the contents of specified memory
location.
X = PEEK (32376)

POINT (x, y)  Tests whether specified graphics cell is on or off.
If horizontal = 0-63; y vertical = 0-31. The value returned is -1
if the cell is in a text character, mode 0 if it is off, or the color code
if it is on. See CLS for color codes.
IF POINT (10, 10) THEN PRINT "ON" ELSE PRINT
"OFF"

POS (device)  Returns current print position. Device 1 =
printer, 2 = display.
PRINT TAB (0$)

PPPOINT (x, y)  Tests whether specified graphics cell is on or off
and returns color code of specified cell.
PPPOINT (13, 25)

RIGHTS (str.length)  Returns right portion of string.
Z = RIGHTS (AD$: 5)

SGN (numeric)  Returns sign of specified numeric expression:
-1 if argument is negative
0 if argument is 0
+1 if argument is positive
X = SGN (A*B)

SIN (numeric)  Returns sine of angle given in radians.
Y = SIN (5)

STRINGS (length, code or string)  Returns a string of
characters (of specified length) specified by ASCII code or by the
first character of the string.
?STRING $(5, "" )$"
?STRING $(5, S$)

STRS (numeric)  Converts a numeric expression to a string.
S$ = STRS (X)

SQR (numeric)  Returns the square root of a number.
Y = SQR (16)

TAN (numeric)  Returns tangent of angle given in radians.
Y = TAN (45)

TIMER  Returns contents or allows setting of timer (0-65535).
?TIMER
TIMER = 0

USRN (numeric)  Calls user's machine-language subroutine.
X = USRN (Y)

VAL (str)  Converts a string to a number.
A = VAL (B$)

VARPTR (var)  Returns address of pointer to the specified
variable.
Y = USR (VARPTR (X))

Control Keys

Cancels last character typed: moves cursor back one space.

SHIFT -Up  Erases current line.
BREAK  Halts program.
CLEAR  Drops everything on screen.

ENTER  Enters a space (blank) character and
moves cursor one space forward.

SPACEBAR  Causes currently executing program to
pause (press any key to continue).

SHIFT  Down  All-caps/upper-lowercase
keyboard switch. (Lowercase
displayed as reversed capitals.)
### Statements

**AUDIO** Connects or disconnects cassette output to TV speaker.

**CLEAR n,b** Reserves n bytes of string storage space. Erases variables. b specifies highest BASIC address.

**CLOAD** Loads specified program file from cassette. If file name is not specified, first file encountered is loaded. File name must be eight character spaces or fewer.

**CLOADM** Loads machine-language program from cassette. An offset address to add the loading address may be specified.

**CLOSE d** Closes open files.

**CLS c** Clears display to specified color c. If color is not specified, green is used.

**FOR TO** Creates a loop in program which the computer must repeat from the first number to the last number you specify.

**GET** (start-), (end-), destination, G, Reads the graphic contents of a rectangle into an array for future use by PUT.

**GOSUB** Calls a subroutine beginning at specified line number.

**GOTO** Jumps to specified line number.

**IF** (THEN) . . . ELSE action 1, action 2 Performs a test.

**INPUT** Causes the computer to stop and wait input from the keyboard.

**LIST** Lists specified line(s) or entire program on screen.

**LINE** (x1,y1)-(x2,y2), PSET or PRESET, BF Draws a line from (x1,y1) to (x2,y2). If (x1,y1) is omitted, the last end point or (128,86) is used. PSET selects foreground color and PRESET selects background color. BF draws a box with (x1,y1) and (x2,y2) as the opposing corners. BF will fill in the box with foreground color.

**LOAD** Loads BASIC program at specified baud.

**NEW** Initializes program environment.

**TITLE** Displays a title on the screen.

**PRINT** Prints alphanumeric characters.

**READ** Reads data from a data file.

**QUIT** Ends program.

**RND** Generates a random number.

**SCHANTZ** Selects or clears screen color.

**SELECT** Assigns value to variable (optional).

**SPACE** Moves cursor n spaces to right.

**ST** Switches to in-screen terminal.

**STOP** Causes the computer to stop execution.

**SUB** Subroutines.

**SLEEP** Causes program to wait for n seconds.

**SPL** Starts printer.

**TAB** Moves cursor (n) spaces to right.

**TICKS** Returns the number of seconds since the computer was turned on.

**TIME** Returns system time.
and EXTENDED COLOR BASIC

LINE INPUT Input line from keyboard.
LINE INPUT ANSWER*4

MID$(oldstr, position, length) Replaces a portion of one
string with another string.
MID$(oldstr, 1, 2) = "KS"

MOTOR Turns cassette ON or OFF.
MOTOR ON
MOTOR OFF

NEW Erases everything in memory.
NEW

ON...GOSUB Multi-way branch to call specified subroutines.
ON...GOTO Multi-way branch to specified lines.

OPEN m.,#d.f Opens file (f) at Screen or Keyboard (O);
Cassette (-1); Printer (-2). For input (I), or output (O).
OPEN "",-1, "DATA"

PAINT(x,y,c,b) Paints graphic screen at point (x,y) with specified color (c) and stopping at border (b) of specified color.
PAINT(10,10,2,4)

PCLEAR a Reserves a number of 1.5 K graphics memory pages.
PCLEAR 8

PCLS c Clears screen with specified color c. If color code is omitted, current background color is used. (See CLS for color codes.)
PCLS 3

PCOPY a Copy graphics from source page to destination page.
PCOPY 0 0

PLAY Plays music of specified note (A-G or 1-12), octave (O),
volume (V), note-length (L), tempo (T), pause (P), and allows execution of subroutines. Also changes (# or +) and flips (-).
PLAY "L1T1A1#P1V1" (L1T1A1#P1V1"

PMODE mode, start-page Selects resolution and first memory page.
PMODE 4,1

POKE (location, value) Puts value (0-255) into specified
memory location.
POKE 15672, 255

RESET Reset a point to background color.
RESET (S,0)

PRINT Prints specified message or number on TV screen.
PRINT "HI"

PRINT# 1 Writes data to cassette.
PRINT# 1
PRINT * A
PRINT# 1,A

PRINT# 2 Prints an item or list of items on the printer.
PRINT # 2,CAPS

PRINT TAB Moves the cursor to specified column position.
PRINT TAB (5) NAME

PRINT USING Prints numbers in specified format.
# Formats numbers.
PRINT USING "##.##", 0.2

D Decimal point.
PRINT USING ".", 0.5

Displays comma to left of every third character.
PRINT USING ",", 0.0

Fills leading spaces with asterisks.
PRINT USING "*", 0.0

Places $ ahead of number.
PRINT USING "$", 0.0

Floating dollar sign.
PRINT USING "$", 0.0

Fixes at position, causes sign to be printed. In last position,
causes sign to be printed after the number.
PRINT USING "###", 5.3

Exponential format.
PRINT USING ".##E###", 0.5

Minus sign after negative numbers.
PRINT USING ".", 0.5

Returns first string character.
PRINT USING "*", 0.0

String field; length of field is number of spaces plus 2.
PRINT USING "Z", 1":BLUE"

PRINT @ location Prints specified message at specified text
screen location.
PRINT @ 255, "HI"
PRINT @ 255, A$?

PSET(x,y,c) Sets a specified point (x,y) to specified color c. If c
is omitted, foreground is used.
PSET(5,6,2)

PUT (start)-(end), source, action Stores graphics from
source onto start-end rectangle on the screen. (Array rectangle
size must match GET rectangle size.)
PUT (3,2)-(5,3), U$PSET

READ Reads the next item in DATA line and assigns it to
specified variable.
READ A$
READ C, B

REM Allows insertion of comment in program line. Everything
after REM is ignored by Computer.
REM THIS IS IGNORED
10 REM THIS IS IGNORED

RENUM newline, startline, increment Allows program line
renumbering.
RENUM 1000,5,100

RESET (x,y) Resets a point.
RESET (14,15)

RESTORE Sets the Computer's pointer back to first item on the
first DATA line.
RESTORE

RETURN Returns the Computer from subroutine to the BASIC
word following GOSUB.
RETURN

RUN Executes a program.
RUN

SCREEN screen-type, color-set Selects either graphics (1) or text (0) screen and color-set (0 or 1).
SCREEN 1,1

SET (x,y,c) Sets a dot at specified text screen location to
specified color.
SET (14,13,3)

SKIPF Skips to next program on cassette tape, or to end of
specified program.
SKIPF "PROGRAM"

SOUND tone, duration Sounds specified tone for specified
duration.
SOUND 128,3

STOP Stops execution of a program.
STOP

TROFF Turns off program tracer.
TROFF

TRON Turns on program tracer.
TRON

Radio Shack
Printed In the U.S.A.
8759101