

BASIC

The *Original -Original* Version

Dartmouth BASIC revolutionized computer programming for the non-experts, who greatly outnumber the experts! It was a simple language, used English words, and gave almost instantaneous response in the days when turnarounds of hours or even days was the norm.

This note gives an outline of the language for those of you familiar with programming.

Statements

There were fifteen statement types in the original BASIC.

LET	Introduces the assignment statement, and is required
PRINT	Provides free-form output
END	Is required
READ	Assigns values to variables from internal data
DATA	Introduces internal data
GOTO	Does just that, transfers to another line-numbered statement
IF	Gives a conditional GOTO
FOR	Introduces the looping construct
NEXT	Terminates the looping construct
GOSUB	Does a GOTO to a subroutine
RETURN	Returns from the end of the subroutine
DEF	Introduces programmer-defined functions
DIM	Allows dimensioning arrays
REM	Provides comments
STOP	Same as reaching the END statement

In addition, the slightly more recent version of BASIC that we are using includes the INPUT statement.

Arithmetic Expressions

Besides the four standard arithmetic operations, BASIC includes raising-to-the-power, the symbol of which is “^”.

Normal precedence rules are used: Exponentiation, multiply and divide, add and subtract. Left-association is used for multiple operations within a group. That is

$(a - b - c)$ is understood to be the same as $((a - b) - c)$

There is one anomaly: a leading minus sign is treated as unary, and has higher precedence than exponentiation, which is contrary to the usual practice. That is

$-x^2$ is understood to be $(-x)^2$

(This was repaired in later versions.)

All arithmetic was done in floating point. In the GE-225 and GE-235, this meant a precision of about 30 bits (roughly ten digits) with an base 2 exponent range of -256 to +255.

Functions

Ten numeric functions were provided.

ABS	The absolute value
ATN	The arctangent
COS	The cosine
EXP	The exponential, i.e., e^x
INT	The integer part (truncating toward 0)
LOG	The natural logarithm
RND	The next random number
SIN	The sine
SQR	The square root
TAN	The tangent

Arguments for SIN, COS, TAN, and the value from the ATN, are assumed to be in radians.

Error messages are given for arguments out of range for the LOG and SQR functions, although SQR then provides the SQR of the absolute value.

The RND function needed a dummy argument to get past the syntax scanner. That is, RND(0). The argument is ignored.

Variables

Variable names can be a single letter, or a single letter followed by a single digit. This provides for 286 possible variable names.

Arrays

A single letter followed by a “(“ denotes an array element, which may be one or two-dimensional. Without a DIM statement, the default dimensions are 0 to 10 for each dimension.

The DIM statement allows other upper limits, but the zero element is always provided.

PRINT Statements

The PRINT statement allows several quantities, including quoted strings, separated by commas (,) or semicolons (;). If by commas, BASIC moves to the start of the next zone. Zones are 15 characters in width. If by semicolons, BASIC does not move but starts the next item at the next space.

Numerical values are printed with either a leading space or a minus sign, and with a trailing space. Thus, numerical values in a PRINT statement with semicolons as separators will have at least one space between values. Furthermore, numeric values will always produce a number of characters that is a multiple of three. Thus,

```
PRINT 12; 34; 56
```

will produce

12 34 56

While there is no string data type, quoted strings are allowed in PRINT statements. If a quoted string and a numeric value are separated by a semicolon in the PRINT statement, the semicolon may be omitted.

If the material on the printed line exceeds 75 characters, an end-of-line is automatically introduced. We sometimes say that the MARGIN is 75.

Defined Functions

The user may define up to 26 new functions, giving them names from FNA to FNZ. Each such function is introduced by the DEF keyword. There must be exactly be one argument. The variable name used as an argument is distinct from the variable with the same name in the rest of the program. The function definition must be a single line with the following form:

```
DEF FNX(X) = <expression>
```

The expression should contain an X unless the function value does not depend upon an argument; the expression may contain other variables from the program.

DEF statement may appear anywhere in the program before the END statement.

Example Programs

```
100 REM  PLOT A NORMAL DISTRIBUTION CURVE
110
120 DEF FNN(X) = EXP(-(X^2/2))/SQR(2*3.14159265)
130
140 FOR X = -2 TO 2 STEP .1
150     LET Y = FNN(X)
160     LET Y = INT(100*Y)
170     FOR Z = 1 TO Y
180         PRINT " ";
190     NEXT Z
200     PRINT "*"
210 NEXT X
220 END
```

```
100 REM  GUESSING GAME
110
120 PRINT "GUESS THE NUMBER BETWEEN 1 AND 100."
130
140 LET X = INT(100*RND(0)+1)
150 LET N = 0
160 PRINT "YOUR GUESS";
170 INPUT G
180 LET N = N+1
190 IF G = X THEN 300
200 IF G < X THEN 250
210 PRINT "TOO LARGE, GUESS AGAIN"
```

```

220 GOTO 160
230
250 PRINT "TOO SMALL, GUESS AGAIN"
260 GOTO 160
270
300 PRINT "YOU GUESSED IT, IN"; N; "TRIES"
310 PRINT "ANOTHER GAME (YES = 1, NO = 0)";
320 INPUT A
330 IF A = 1 THEN 140
340 PRINT "THANKS FOR PLAYING"
350 END

```

Commands

Although not part of BASIC, the commands of the operating system include the following:

HELLO	Start a new session, enter your user number
NEW	Start a new program
OLD	Retrieve a program from storage
SAVE	Save the current program to storage
REPLACE	Save the current program to storage, overwriting older version
RENAME	Rename the current program
CAT	List the names of your saved programs (short for CATALOG)
LIST	List the current program
RUN	Run the current program
STOP	Stop the current run of the program (in case an infinite loop)
UNSAVE	Unsave the current program program
SYSTEM	Name the system -- limited to either BASIC (default) or ALGOL
BYE	End the session
GOODBYE	Same as BYE

All commands may be abbreviated to the first three letters.

The NEW, OLD, and RENAME commands may be followed by a program name. If not, the operating system will ask you for the name of the program. The SYSTEM command may be followed by either BASIC or ALGOL. If not, the operating system will ask you for a system name. (Like the commands, the system names may be abbreviated to three letters.)

In addition, the SPEED command allows you to specify the teletype speed, for a more realistic simulation. Thus, SPEED 10 will slow things down to about 10 characters per second.

See the separate pdf document *Commands.pdf*.

Disclaimer

This brief manual describes the version of BASIC included in the emulation. The line spacing, and the response "READY", as provided by the Datanet-30, may not be quite correct. DTSS also supports Algol; see the separate pdf document *An Algol Outline.pdf*.