Intro to Computing Lab
Paul L. Bailey
Laboratory III - BASIC Interpreter

Introduction

At this point, you have made a DOS prompt, downloaded some utilities, and gained some familiarity with executing commands from the DOS prompt.

Among the files in `C:\UTIL` are `GWBASIC.EXE` and `GWHELP.CHM`.

Open My Computer, navigate to the `C:\UTIL` folder, and double-click on `GWHELP`. This will open a window which contains the entire Microsoft manual for GWBASIC. Peruse this for a moment, then minimize it. Keep it handy. We are about to begin programming in BASIC.

Being an interpreter, BASIC may be used in two modes:

- **Direct Mode:** Type commands or instructions to the interpreter, to which it immediately responds.
- **Indirect Mode:** Create a program of commands, to be run as a set.

Direct Mode Demo

Click on the DOS icon and type `CD TEST`. This will put you in the `TEST` directory. We will make some BASIC programs here; but first, we try out a few commands.

Type `GWBASIC`. This opens the GWBASIC interpreter. You should see the `Ok` prompt. Enter the following commands.

```
PRINT "TEST"            Prints the string "TEST", without the quotes.
PRINT TEST              Prints 0, because TEST is a variable; it is a symbol
                        which represents a value.  The value can be changed by
                        the LET command.
LET A=2                 Sets the variable A to the value 2.
                        This 2 is stored in floating point format.
LET B=3                 Sets the floating point variable B to the value 3.
PRINT A+B               Evaluates A+B to be 5, and prints 5.
LET S="TEST"            Displays the error message "Type mismatch", because A is
                        a floating point variable.  The variable left unchanged.
LET S$="TEST"           Sets the string variable A$ to the value "TEST".
PRINT S$                Displays the string "TEST", without the quotes.
LET C=A/B               Sets the floating point variable C to approximately 2/3.
PRINT C#                Prints .6666667.
LET C%=A/B              Sets the integer variable C% to 2/3, rounded to an integer.
PRINT C%                Prints 1.
INPUT A#                Prints ?, reads from the keyboard, and sets the
                        integer A# to the value you typed.  If you do not type
                        and integer, the BASIC prints "?Redo from start",
                        and asks again for an integer.
PRINT A#;               Prints the integer you typed.
INPUT T$                Prints ?, reads from the keyboard, and sets the
                        string variable T$ to the value you typed.  You can
                        type anything, because the input is interpreted as a string.
PRINT T$                Prints the string you typed.
```

This demonstrates several things:

- You can type commands directly to the system, and get immediate feedback.
- We have seen the command `PRINT`, `LET`, and `INPUT`.
- We have seen the variable types floating point, integer, and string.

Indirect Mode

When a number precedes a statement, the statement is not immediately executed; instead, the entire line (number + statement) becomes part of the current program. The number is called the *line number.* When the program is run, the statements in it are executed consecutively, according to their line numbers.

Type the following; this will enter and save a new program.

```
NEW                      Clears the current program and all variables; start fresh.
10 REM <your name>       REM means REMARK; this line does nothing but inform the reader.
20 PRINT "ADDITION"      The system automatically produces a carriage return at the end
                         of a PRINT statement.
30 PRINT "A";            The semicolon at the end of the PRINT statement suppresses the line feed.
40 INPUT A               A question mark will appear after the "A" you just printed.
50 PRINT "B";            Without the semicolon, the system would go to the next line after "B".
60 INPUT B
70 PRINT "A+B =";A+B     Prints the string "A+B=", followed by the value of A+B.
SAVE "ADD"               Saves the program in the file "ADD.BAS" in token format.
DELETE                   Erases the program.
LOAD "ADD"               Loads the program.
LIST                     Lists the program.
RUN                      Runs the program.
SAVE "ADD",A             Saves the program in the file "ADD.BAS" in ASCII format.
SYSTEM                   Returns to DOS.
TYPE "ADD.BAS"           You see that the program is now in a text file.
```

Colors

In the basic interpreter, type the following commands.

```
COLOR 10                 Sets the foreground color to bright green (10)
COLOR 14,1               Sets the foreground color to bright yellow (14) on blue (1)
CLS                      Clears the screen.
```

The foreground and background colors are stored in 8 bits, each using 4 bits. The bits are 1=blue, 2=green, 4=red, 8=bright. Each of the four bits can be on or off, and they give 8 combinations of color and two possible intensities.

```
  0 Black                4 Red                     8 Dark Grey      12 Bright Red
  1 Blue                 5 Magenta = Blue+Red      9 Bright Blue    13 Bright Magenta
  2 Green                6 Brown = Green+Red      10 Bright Green   14 Bright Yellow
  3 Cyan = Blue+Green    7 Grey = Blue+Green+Red  11 Bright Cyan    15 White
```

This program displays all foreground colors. Type it in, run it, and save it in ASCII format as `COLOR.BAS`

```
10 REM Display Colors  Paul L. Bailey
20 COLOR 0,0:CLS
30 LET F%=0
40 IF F%>=16 THEN END
50 COLOR F%,0
60 PRINT "Color";F%
70 LET F%=F%+1
80 GOTO 40
```