A REVIEW OF THREE STRUCTURED BASIC PACKAGES:

True BASIC 2.0
Turbo BASIC 1.0
Microsoft QuickBASIC 3.0

True BASIC (2.0) True BASIC, Inc. 39 South Main Street Hanover, NH 03755; (800) TR-BASIC.

Turbo BASIC (1.0) Borland International 4585 Scotts Valley Drive Scotts Valley, CA 95066; (408) 438-8400.

QuickBASIC (3.0) Microsoft Corp. 16011 NE 36th Way Redmond, WA 98073; (800) 426-9400.

The question of whether or not to use BASIC as the language for a first course in programming (usually for non-majors) has become increasingly controversial over the past five or ten years. Many computer science and mathematics faculty who teach such a course are frustrated by the difficulty, due to the language's lack of advanced control structures, of using BASIC to implement the principles of structured programming. The demand for these courses persists, however, because of the widespread use of BASIC in the social and physical sciences.

In January of this year, the American National Standards Institute (ANSI) approved a new standard for BASIC which fits better with current notions of what constitutes a good programming language. Several software companies offer products that conform closely to the new standard and that offer other advantages, such as an easy to use programming environment. The availability of a structured BASIC packaged with a convenient interface may well be the solution to the problems that the older versions of BASIC presented.

In this review, three such products are examined. Language features, interface, manuals and texts, and costs are described and compared. The review restricts itself to those product features that might be important in a freshman level course (so, for example, the ease of interface with assembly language is not discussed).

The three packages considered are True BASIC 2.0, Turbo BASIC 1.0, and Microsoft QuickBASIC 3.0, all for the IBM PC and compatibles (True BASIC is also available for the Macintosh, Amiga, and Atari ST). These products all provide environments in which programs can be edited, compiled, run, and debugged. They were chosen over other software because of the reputations of the companies. True BASIC, Inc., is headed by John G. Kemeny and Thomas E. Kurtz, the original developers of BASIC in the 1960's. Kurtz chaired the ANSI committee that proposed the new standard in 1984, and True BASIC debuted later that year. Turbo BASIC is a product of Borland International, whose earlier product Turbo Pascal has been an enormous success. Microsoft Corporation is a major producer of software for IBM; for example, they wrote the operating system for the IBM PC series (as well as for the new PS/2 series) and IBM BASICA.

Language Features

Control Structures

All three products contain the following control structures which essentially eliminate the need for GOTO's. I've noted differences between the three, as well as differences from older versions of BASIC, such as VAX-11 BASIC and IBM BASICA:

1. Block IF-THEN-ELSE statements which allow tests of multiple conditions. For example,

```basica
if age < 12 then
    print "NO CHARGE"
else if age < 18 then
    print "$2.00, please."
else if age < 65 then
    print "$5.00, please."
else if age > 65
    print "$3.00, please."
end if
```

Older versions of BASIC such as IBM BASICA have a more restricted IF-THEN-ELSE which requires nesting to implement the above example. Note, by the way, the in-line comment after the last "else". QuickBASIC and Turbo BASIC use a single quote to start a comment (as does IBM BASICA), while True BASIC (like VAX-11 BASIC) uses an exclamation point instead. All three products retain the alternative REM form of a comment.

2. A SELECT-CASE statement. With a few exceptions (such as VAX-11 BASIC), this useful construct is new to BASIC. The first example could be rewritten using a SELECT-CASE structure. Another such example is:

```basica
select case score 'an exam score
    case 10
        print "EXCELLENT"
    case 8, 9
        print "VERY GOOD"
    case 5 to 7
        print "POOR BUT PASSING"
    case is < 5
        print "FAILING"
end select
```

3. An extremely flexible DO-LOOP structure. This is really four loop structures in one, since you can add "WHILE expression" or "UNTIL expression" at either the beginning or end of the loop. For example,

```basica
input value
do until value = -1
    count = count + 1
    total = total + value
input value
loop
```

The question of whether or not to use BASIC as the language for a first course in programming (usually for non-majors) has become increasingly controversial over the past five or ten years. Many computer science and mathematics faculty who teach such a course are frustrated by the difficulty, due to the language's lack of advanced control structures, of using BASIC to implement the principles of structured programming.
True BASIC actually allows a "WHILE expression" test at both ends of the loop, i.e. DO WHILE expression1 ... LOOPWHILE expression2.

4. Alternative EXIT statements for the DO-LOOP structure and the FOR-NEXT loops, e.g.

input x
reply$ = "ITEM NOT FOUND"
for i = 1 to n
if x = numList(i) then
  reply$ = "ITEM FOUND"
  exit for
next i

All three products also have EXIT DEF and EXIT SUB statements. In addition, Turbo BASIC has EXIT SELECT, EXIT IF, and EXIT WHILE (for the WHILE .. WEND structure). True BASIC has EXIT PICTURE ("Picture" is a True BASIC graphics statement, discussed later in the review).

Functions, Subprograms, and Modularity

All three products improve greatly on earlier BASICS in these areas. Multi-line function definitions are permitted, in contrast to most older BASICS (VAX-11 BASIC is one exception). The syntax and other features differ somewhat in each case. First, True BASIC function definitions do not allow the possibility of local variables. QuickBASIC allows "static" variables, which exist only within the function (or subprogram), but which, unlike local variables, are not re-initialized on each call of the function. Turbo BASIC is the most flexible here, permitting local, static, and global variables.

Another important difference is that both True BASIC and Turbo BASIC support recursion in functions and subprograms, while QuickBASIC does not. Here is an example of a multi-line function definition. Note that Turbo BASIC and QuickBASIC require that the function name begin with "FN", while True BASIC does not.

```basic
def FnChoose (n, r)
  if n < 0 or n < r then
    FnChoose = 0
  end def
  top = 1
  for i = n-r+1 to n
    top = i * top
  next i
  bot = 1
  for i = 1 to r
    bot = i * bot
  next i
  FnChoose = top / bot
end def
```

As one would expect, these new BASIC packages also have a flexible subroutine structure that allows the passing of parameters (QuickBASIC uses the word "subprogram" to distinguish it from the old GOSUB ... RETURN construct, which all three retain). In all three, parameters may be passed by reference (the default) or by value (signaled in the call to the subprogram by enclosing the argument in parentheses). As mentioned above, QuickBASIC does not support recursion, and each subprogram definition must end with the word "STATIC" to indicate that it is non-recursive. The syntax for listing parameters is slightly different for each product. As an example, here is a routine to bubble-sort a list of numbers:

```basic
rem This routine will run in Turbo BASIC or QuickBASIC.
rem The word "static" in the subprogram heading is
rem required in QuickBASIC, optional in Turbo BASIC.
sub BubbleSort (numList#, length%)
  for i = 1 to length%-1
    for j = i+1 to length%
      if numList# (j) < numList# (i) then
        temp = numList# (i)
        numList# (i) = numList# (j)
        numList# (j) = temp
      end if
  next j
  next i
endsub
```

The "numList# (1)" notation in the parameter list indicates that the first argument will be a 1-dimensional array. The suffix '#*' on the variable name "numList" indicates that the array entries are single-precision and the "%" suffix on "length" signals a variable of type integer. This use of suffixes to declare the types of parameters is required in QuickBASIC, optional in TurboBASIC, and forbidden in True BASIC. In addition, the syntax for array parameters is slightly different in True BASIC. The call to this routine (in Turbo BASIC or QuickBASIC) would look like:

```basic
call BubbleSort (numList#, length%).
```

Both Turbo BASIC and QuickBASIC have the metastatement "$include", which will cause separately stored functions and subroutines to be compiled as part of the current program. QuickBASIC and True BASIC also provide for "user libraries", i.e., collections of executable code files which can be called by individual programs). Finally, True BASIC has modules, which allow more choices for the sharing of data between the external module and the calling program.

Sound and Graphics

All three packages have the sound and graphics capabilities of IBM BASICA plus some extras. Each of the three supports systems having monochrome, CGA, or EGA adapters and offers the appropriate choices of screen resolutions. QuickBASIC and Turbo BASIC have retained (with some syntactical changes) the following IBM BASICA graphics statements: CIRCLE, COLOR, DRAW, GET, LINE, PAINT, PMAP, POINT, PRESET, PSET, PUT, SCREEN, VIEW, and WINDOW. In addition, these two packages have added a PALETTE statement, to be used with the IBM EGA adapter, which allows the palette colors to be changed. True BASIC
has redesigned its set of graphics statements to make them easier to use and more machine independent. For example, English names of colors ("red", "blue", etc.) are used in statements to set foreground and background colors. True BASIC also introduces a graphics analogue to the subroutine structure called PICTURE which is called using a DRAW statement. Also, the call can be accompanied by a sequence of "transformations" (shift the picture, change its scale, rotate it, shear it).

For example, if a picture EqTriangle is defined, which draws a specific equilateral triangle, then the following statement will shift it to the right and rotate it 45 degrees: draw EqTriangle with shift (0.5) * rotate (pi/4). This construct allows the user to do some nice graphics with relatively simple syntax.

Line Numbers and Labels

As the examples in the preceding sections demonstrate, all three BASIC packages have done away with line numbers. The one exception here is that in True BASIC, line numbers must be used with GOTO and GOSUB statements. QuickBASIC and Turbo BASIC use alphanumeric line labels instead.

Compiled vs. Interpreted Code

Last, but not least, all three languages are compiled rather than interpreted (although True BASIC can also behave as an interpreter). The compilation time is so fast, however, that students are still able to get the same kind of immediate feedback during program development that helped make BASIC attractive as a first language to begin with. Moreover, using a compiled language emphasizes the difference between a syntax error and a runtime error. In addition, the final executable code runs much faster than interpreted code.

Interface

Programming Environment

All three software packages provide a complete environment for program development, combining full-screen editor, compiler, run-time system, and varying degrees of debugging facilities. QuickBASIC and Turbo BASIC are "menu-driven" (although some aspects of Turbo BASIC are command-driven, as explained below), while True BASIC is "command-driven" (however, the versions of True BASIC for the Macintosh, Amiga and AtariST are all menu-driven).

The True BASIC programming environment consists of two areas or "windows": the editing window on top is the work area for the program text, while the lower window is where the user enters commands to load and save files, compile and run programs, etc. True BASIC calls this the History window (because the user can scroll backward to see commands used earlier in the editing session). There is good on-line help, but a novice user will have to have at least some minimal documentation to get started. For example, one must know to type "OLD filename" to load a file. Similarly, one must know to type "BYE" to quit gracefully.

The Turbo BASIC environment uses four windows and a pull-down menu system with headings across the top of the screen. The four windows are Edit (the work area), Message (displaying information about the state of the current process, e.g., what line is being compiled, etc.), Run (where the output of the program is shown), and Trace (for tracing and debugging information). The main menu headings are File, Edit, Run, Compile, Options, Setup, Window, and Debug. The default screen, divided into four windows plus the menu headings, can be a bit overwhelming to a novice user. TurboBASIC provides a "Zoom" option, activated by a function key, that makes the current window fill the screen. While most simple editing tasks can be accomplished via the menu system, TurboBASIC resorts to some Wordstar commands for more advanced editing. For example, to search and replace a text segment, the user must type ctrl-QA. Documentation is a must to use these editing commands.

Of the three products, QuickBASIC is the only one that is almost fully menu-driven. The initial QuickBASIC screen has a list of menu headings across the top (File, Edit, View, Search, and Run) and the rest of the screen is the work area for the program text. When first started, an information screen explaining how to use the menus is displayed. Of course, it takes a little practice for a novice user to become comfortable with the menus, but in a short time, and largely without additional documentation, one is able to easily access most of the editing features. For students in a first course in programming, the QuickBASIC environment is certainly the friendliest of the three.

Tracing and Debugging

These products have somewhat different facilities for tracing and debugging programs. TrueBASIC has no separate debugging/tracing program per se, but the user can set breakpoints and then use them in conjunction with the interpreter mode to inspect variables. For example, if a program has a loop in which variable i changes, the user can set a breakpoint in the loop and run the program. When the running program pauses at the breakpoint, typing "print i" in the history window will show the current value of i.

Turbo BASIC has a tracing option which can be set on or off. When set, the trace window displays line numbers, labels, and procedure/function names, and the user can toggle the running program between single-step and execute modes. While in tracing mode, the user can single-step to the next line number or label; note that one cannot single-step to a program line that is not labeled or numbered, and so many programs cannot be traced at all with this debugger.

QuickBASIC provides by far the most extensive tracing and debugging facilities of the three. When the debugger is activated, the screen is split into two areas, with output at the top of the screen and debugging information at the bottom. The user can adjust how many lines of program code are shown in the debugging area, and has the option of various levels of tracing. One can single-step, either following program execution into func-
tion and procedure calls or treating calls as a single statement; one can "animate" the program, i.e. execute the program slowly while highlighting each line as it is reached; finally, one can simply switch to regular program execution without tracing or animation. The user can also set multiple breakpoints and may choose one "watch" variable - a program variable whose contents are displayed throughout tracing.

Manuals and Texts

Turbo BASIC comes with an "Owner's Handbook" which combines a user guide to the programming environment and the language with an alphabetical reference directory. Microsoft QuickBASIC does essentially the same thing. Both books are useful for reference, but neither could possibly pass as a text for an introductory course. I found the explanations in the QuickBASIC manual to be somewhat clearer than those in the TurboBASIC book. On the other hand, one irritating feature of the QuickBASIC manual is that, when updating the software from version 2.0 to version 3.0, Microsoft did not revise the manual; instead they added a 70 page update section, leaving the remaining 595 pages sprinkled with some confusing inaccuracies.

True BASIC comes with a user's guide, a reference manual, and a Version 2.0 supplement. The user's guide is a collection of tutorials designed to be used in conjunction with the many sample programs supplied on the True BASIC disk. It could be used as a textbook, but it has no exercises, and in any case there are quite a few textbooks available for True BASIC. The reference manual covers the language and programming environment in greater detail. It is organized into textbook-style chapters according to topic, rather than alphabetically. Finally, the Version 2.0 supplement is a booklet describing changes from True BASIC 1.0; as is the case with the QuickBASIC manual, the True BASIC user's guide and reference manual have not been revised.

Texts

The amount of textbook support for these products may change rapidly, but as of this writing the situation is feast (True BASIC or Turbo BASIC) or famine (QuickBASIC and Turbo BASIC). There are at least seven or eight texts available that use True BASIC and at least three of these are suitable for use in an introductory course. On the other hand, there are no introductory level texts available for QuickBASIC or Turbo BASIC. There is one QuickBASIC text (Microsoft QuickBASIC by Douglas Hergert) available from Microsoft Press, but it is written for experienced BASIC programmers. For example, after introducing the QuickBASIC programming environment in Chapter 1, the topic of Chapter 2 is "Modular Programming: Subprograms and User-Defined Functions" - certainly not an appropriate first language topic for an introductory level course. Instructors of introductory courses might consider using one of the True BASIC texts, regardless of which software they buy.

Costs

All three of these products boast low price tags, with even lower costs for educational users. Discounts of 25-50% are available from various mail-order companies, which may be of interest to buyers who don't wish to meet the minimum purchase requirements necessary to get the manufacturers' educational prices. The manufacturers' prices, in order of decreasing cost, are described below.

True BASIC's list price is $150, but with a $1000 minimum purchase (in the first year; the minimum purchase amount decreases in subsequent years), the price drops to $50. Buyers who meet the minimum purchase requirement may also buy partial packages: the software and either manual for $36, either manual for $20.

Turbo BASIC's list price is $100, but to its price to educational institutions (under the "Borland Scholar Program") is $40. In addition, if ten or more copies are purchased, the price is reduced to $30. QuickBASIC's list price is $99 and there is an educational discount of 45% with a minimum purchase of $1000, reducing the price to about $55. If the software is to be used in a microcomputer laboratory setting with a minimum of ten machines (there is no $1000 minimum for this "lab pricing"), then the price drops to $25.

Summary and Recommendations

True BASIC

Of the three products, True BASIC offers the richest language. It has recursion (QuickBASIC does not), and it has graphics structures that are far superior to those of the other two. Another True BASIC language feature not mentioned above is a collection of "MAT" statements for convenient handling of matrices, including INPUT, READ, WRITE, PRINT, AND PLOT (a single statement for graphing an entire array of points); neither TurboBASIC nor QuickBASIC have MAT statements. It is the only product of the three that has an interpreter in addition to the compiler and the only one that has modules. My main objection to TrueBASIC is its programming environment. The menu-driven environments of the other two products are vastly easier and more pleasant to use than True BASIC's split-screen, command-driven environment. On the other hand, True BASIC is the only product available for machines other than IBM PC compatibles and its versions for the Macintosh, Amiga, and Atari ST are menu-driven.

Turbo BASIC

Turbo BASIC, like True BASIC, has recursion; it is the only language of the three to have true local variables (as opposed to the static variables of QuickBASIC) for subprograms. With its multi-window, menu-driven environment, Turbo BASIC is much easier to use than True BASIC, but some users may find the multiple windows confusing rather than
helpful. As mentioned above, a single window environment can be obtained using the "Zoom" option; it is also possible to customize the size and arrangement of the windows to the user's liking. Turbo BASIC also has good on-line help; the user can get a help screen for any highlighted menu option. I did not find Turbo BASIC's tracing/debugging facilities to be very useful, limited as they are to tracing only labeled program lines and calls to procedures or functions.

Microsoft QuickBASIC

QuickBASIC does not have some of the language features of True BASIC or Turbo BASIC; it does not have recursion and it does not have the exceptional set of graphics routines that True BASIC possesses. However, these may not be of primary importance if the language is to be used in an introductory level course. Like the other two products, it has a good set of control structures (IF..THEN, SELECT..CASE, FOR..NEXT, DO..LOOP), and its programming environment is the nicest of the three. In addition, it is the only product that has a really useful tracing and debugging facility.

Recommendations

These are all excellent products which are major improvements on earlier versions of BASIC. Which product you choose depends upon your particular needs. Any instructor who wishes to emphasize graphics should seriously consider True BASIC. A colleague of mine plans to use it in an interdisciplinary course on symmetry and the work of M.C. Escher. Of course, if your BASIC course will not be taught on IBM PC's, True BASIC is the only choice (it is also the only choice if you want a textbook written specifically for that language). If IBM PC's are to be used and graphics are not a major consideration, then the menu-driven programming environments of Turbo BASIC and QuickBASIC may make them the products to consider. If the availability of recursion is important, then consider TurboBASIC; otherwise, the better programming environment and excellent debugger of QuickBASIC may make it the best choice.

This review grew out of work I did on a department subcommittee that chose new BASIC software to use in our introductory programming course for nonmajors. After reviewing these three products, we decided to use Microsoft QuickBASIC, together with an introductory text written for True BASIC - not a perfect solution, but the one that seemed best for our purposes. The software has already been used successfully in a summer course. With any of these products, however, we were all confident that the course would be greatly improved because all three variants of the language allow the student to gracefully implement the principles of top-down design and structured programming.

Reviewed by Alice M. Dean, Department of Mathematics and Computer Science, Skidmore College, Saratoga Springs, NY 12866.