The New Nation

Learn BASIC and run your computer

Naser Siddiqui

BASIC works well when there are loops. Loops are indented and programs are created to get the job done. The speed of BASIC is slower than some languages. But that does not matter in many situations.

For example, if you are calculating a mortgage payment, BASIC might take three seconds in comparison to one third second in a faster language. In many other cases the speed is not any problem but the algoruthm used needs due attention. Let a bubble sort that alphabetizes a list of 500 name night run very slowly. You could rewite the program in another language and that would run the bubble sort faster. But if you changed to a shell sort or a quick sort, you would notice a definite improvement in speed, even in BASIC also. There exist many techniques for speeding up BASIC programs.

To make comparison fair, BASIC programs should be compiled or in other words run through a special process which translates them into something resembling machine lauguage. A compiled BASIC program often runs at the same speed as a similar program in another language. Using a compiler one can have the best of both worlds.

The computer executes BASIC program lines in numerical order unless the program instructs the computer to do otherwise. There are several ways to transfer control to a different lime, one of those ways is looping. One of the salient aspects of the looping is the use of GOTO command.

GOTO tells the computer to go to another line specified by a line number or to a line label in Amiga BASIC and Atari BASIC. For example: 150 GOTO 390 (transfer to line 390) 550 GOTO WHEELS (transfer to line labeled WHEELS).

You can transfer to a previous line, a laterline, or even the same line. A toop is created when the computer executes a statement or several statements repeated lf. If you are in the middle of a GOTO loop, you can stop the computer by pressing a break bey or key combination quite often control-C ;
press Run/Stop on the key board of the computers.

Now let us see an example of a program using GO TO statements to after the normal flow of a program :

```
20  PRINT "ONE"
30  GOTO 60
40  PRINT "TWO"
50  GOTO 80
60  PRINT "THREE"
70  GOTO 40
80  PRINT "FOUR"
90  GOTO 90
100 END
```

Instead of printing the words One, Two, Three, Four in order as they appear in the program, the computer moves around. First the word ONE is printed, then the computer is instructed to GOTO line 60, which prints THREE. Next the computer executes GOTO 40, which branches back to line 40 where TWO is printed. Then the computer finds GOTO 80, which prints the word FOUR. Line 90 says GOTO 90, which means the computer will repeatedly execute line 90—it will be stuck up in a loop.

But you do not actually see anything happening on the screen, but the program will not end, so you know you are in a loop and need to break out of the program to regain control of the computer.

Sometimes this type of loop is helpful to keep some material like graphics on the screen. Another type of GOTO loop could be used which is as follows :

```
20 PRINT "HELLO"
30 GOTO 20
```

Here is a GOTO loop using a variable :

```
20 A—A—I
30 PRINT A
40 GOTO 20
```

Thus GOTO command plays a vital role in BASIC.