

5. LOOPS

FOR NEXT STEP

FOR NEXT is the main looping technique in most BASICs. Advan BASIC allows two kinds of FOR loops, integer and real. The name of the loop variable determines the loop type. The lower limit, upper limit, and step can be integer or real, whether or not the FOR loop is integer or real. If there are differences in type, the system will convert so that the lower limit, upper limit and step are of the same type as the loop variable.

In some BASICs you do not need a variable name with the NEXT statement. In Advan BASIC you must include the variable name with each NEXT, and the system will check to make sure that corresponding FOR and NEXT statements have the same variable name. As in most BASICs, FOR NEXT loops may be nested. Here a real loop is nested inside an integer loop:

```
10 FOR T%=1% TO 5%
20   FOR Y=1 TO 3
30     PRINT T%,Y
40   NEXT Y
50 NEXT T%
```

And here an integer loop is nested inside a real loop:

```
10 FOR T=5% TO 1% STEP -1
20   FOR Y%=1 TO 3 STEP 2
30     PRINT T,Y%
40   NEXT Y%
50 NEXT T
```

The 5% and 1% in line 10 will be converted to real numbers, and the 1, 3, and 2 of line 20 will be converted to integers.

If you are writing programs to move data around inside the computer, you generally want to work with integers because they are faster. Remember, the largest value of an integer is 32767. Advan BASIC however, will let you use FOR loops and POKEs with integers up to 65535%. It simply stores them as negative numbers. Consider the following program, which stores zeroes in the bottom line of a mode 0 text display (memory locations 40920 to 40959):

```
10 FOR T%=40920% TO 40959%
20   POKE T%,0%
30 NEXT T%
```

As far as the program is concerned, it is executing a loop from -24617% (same as 40920%) to -24577% (same as 40959%).

WHILE WEND

If you know how many times you are going to repeat a loop, FOR NEXT statements work very well; if you do not, FOR NEXT is not so easy to use. For example, suppose you write a program in which the user enters numbers to be summed, with -1 entered after the last number. Here you have no idea how many numbers will be entered. A WHILE WEND or REPEAT UNTIL loop,