CIRCULAR MEASURE

We look at 'recursion', a technique used in advanced programming such as artificial intelligence and the writing of compilers and assemblers. A functional knowledge of recursion can enhance a programmer's skills and add new dimensions to your BASIC programs. Our simple example, a Towers of Hanoi game, shows how easily the technique can be used.

The subject of this investigation is best summed up by a common joke definition:

Recursion: *see* Recursion

This circular definition demonstrates one essential feature of recursion — namely, something being defined in terms of itself. But it ignores another important feature: for recursion to be workable, there must be a way out of the circularity.

The puzzle we have used to illustrate recursion is The Towers of Hanoi. The puzzle begins with a pile of discs arranged in order of size, with the largest disc on the bottom of the pile and the smallest disc on top. To solve the puzzle, you must move all of the discs from the first pile to a second pile according to the following rules:

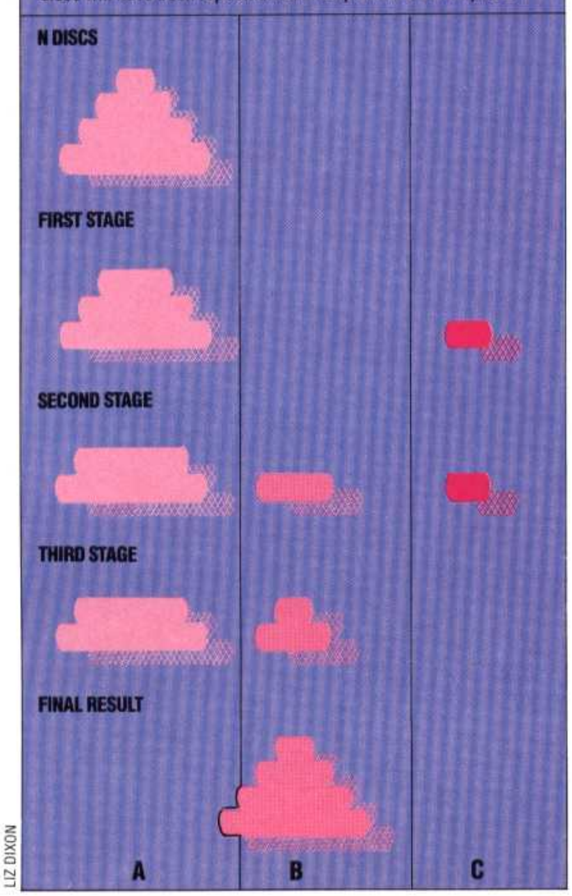
1) Only one disc may be moved at a time;
2) A disc may not be placed on a smaller disc;
3) There may never be more than three piles of discs.

The diagram illustrates how we utilise the concept of recursion to make the problem manageable. We begin with a pile of four discs. By assigning a variable N with the value of four, we indicate the total number of discs that must be moved. Since the rules do not allow the movement of more than one disc, we use a recursive formula to reduce the value of N by 1, then continue the calculation until N equals one. When N = 1, the program stops calculating and moves the appropriate disc.

If we are working with a version of BASIC that allows recursion, it is easy to write a program that follows the above process exactly. In the BBC BASIC program, all the work of calculating the moves is done in lines 1000 to 1050. The rest of the program is required to produce the moving pictorial display!

Tower Power

The program asks the player to input the number of discs to be moved by typing in the required number. The assigned number of discs are placed in pile A. The recursive algorithm then moves one disc at a time through piles A, B and C, in the first three stages shown below. Eventually, the original pile of N discs is reduced to zero and there are no more discs to move, as shown in the final result. When the movements have been completed, N discs will have been repositioned in sequential order in pile B



LIZ DIXON

THE SPECTRUM VERSION

To convert the Towers of Hanoi program to Spectrum BASIC we have to replace a recursive procedure with a recursive subroutine, which begins at line 1000 of our listing. Each time the subroutine has to make a recursive call to arrays M, A, B, or C, it increments the pointer variable J and puts the new variable values into M(J), A(J), B(J), and C(J). Subsequently, these new values can be used in the next call to the subroutine without disturbing the old values. At the end of the subroutine, the value of J is decremented, thus restoring the old values. This method can always be used for writing recursive subroutines in BASIC, no matter how complicated the recursion.

The display section of the program is straightforward, printing an object in a new position and erasing it by printing blank characters in the old position. The programs show the side view of a pile of discs. To make the piles look symmetrical, we have ended each odd-sized bar with graphics characters half made up of a space and half solid colour.