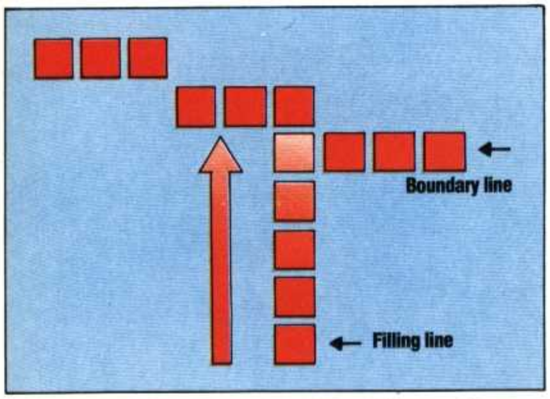
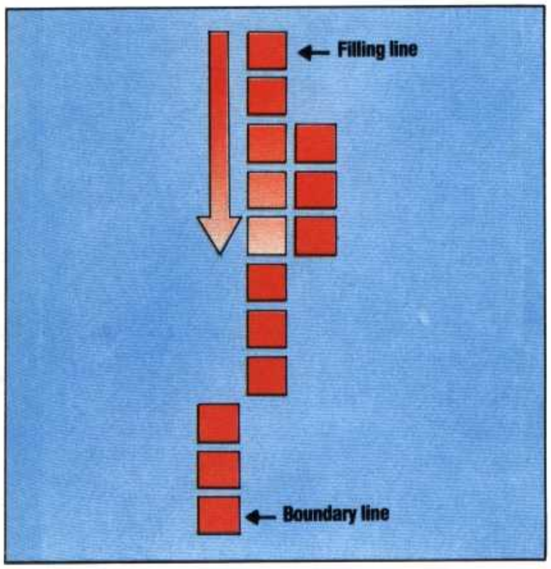
PROBLEMS WITH GRADIENTS

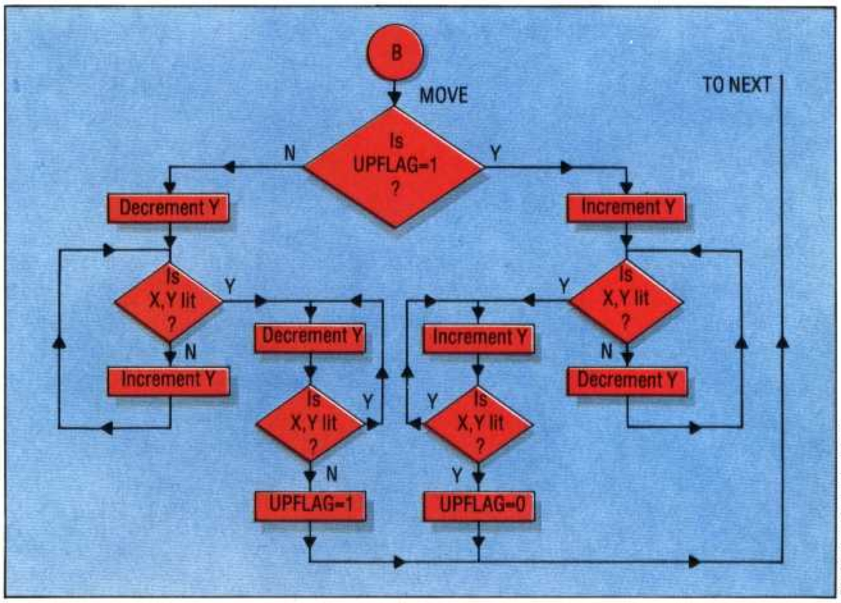
The algorithm to cope with all the different possible types of line — shallow, steep, thick and thin — looks fairly complex, but the principles of its operation are relatively simple. Let us consider the situation where the fill routine is moving up the screen, and it encounters a line with a fairly shallow gradient.



The first action that the algorithm takes on encountering the line is to backtrack one pixel. If the move is to the right then a further move needs to be made so that the next fill line can start at an unlit position. If we look at the case where a downward-moving fill line encounters a steep line we can see another problem to overcome.



If the fill is to move right then it must backtrack three pixels to find an empty space from which to start its next upward fill. If, however, the fill is to move left, then it must move another three pixels down the screen so that the new upward fill starts in the first unlit position above the boundary line. For each fill direction we must therefore have two loops to move in either direction until the first unlit position on the fill line is located. The flowchart showing this algorithm is given above.



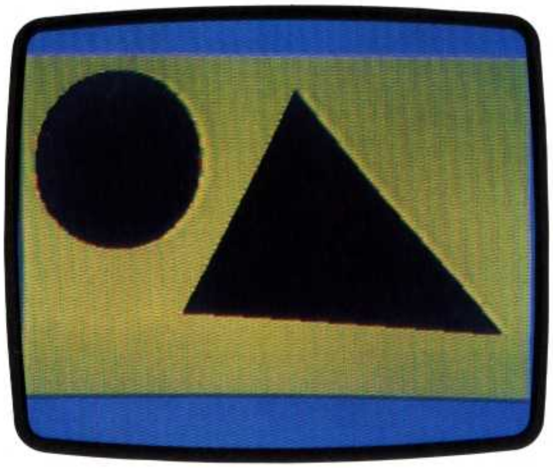
DIAGRAMS BY KEVIN JONES

THE FILLSUB LISTING

The main body of Fillsub follows the flowchart we have outlined fairly closely. Labels that have been used in the source code listing have been included in the corresponding positions on the flowchart to help you find your way around the program. In several places the fill algorithm requires us to test whether a point is lit or not. The program does this using a subroutine called POINT, which calculates the address of the point from its X,Y co-ordinates in the same way that our Plotsub program does. But instead of performing a logical OR to set a bit to one, POINT uses logical AND to see whether the bit in question is one or not. If the result of the AND operation is not zero, then the bit is a one. This result is stored in PTFLAG for later testing by the main program. Much of POINT does the same work as Plotsub and readers who feel comfortable with 6502 machine code may wish to modify Plotsub so that it can not only turn on a bit in the bitmap, but also test the value of any bit.

Finally, let's look at how the Fillsub routine is used. First of all, Fillsub needs several parameters to be passed to it. These are:

- The co-ordinates of the start point. This *must*be inside the shape we want to fill!
- The maximum and minimum X co-ordinate values to be used. Care must be taken when dealing with shapes that contain very acute angles, because Fillsub can move outside a shape if it reaches the edge of the interior space before



Shaping Up
After a number of articles, and having typed several kilobytes of code, we achieve finally on the Commodore 64 the kind of elementary hi-res graphic facilities that are taken for granted as standard BASIC commands on almost every other home micro on the market. The alternative to this kind of effort, of course, is to spend £30 to £50 on an extended BASIC cartridge