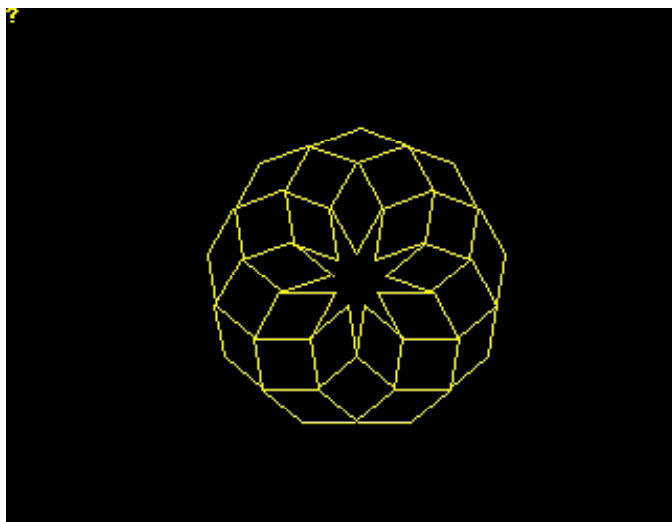


## Tuttle - a screen turtle

Don't let the shortness of this program fool you; many people will consider this to be the most exciting and creative program in the book and perhaps the most creative that they have seen for a long time. Briefly stated, it draws patterns on command. Having said that, however, we are a long way from realising its potential. Consider a polygon - any polygon - with any number of sides from three (a triangle) to say 30 (which will appear as a circle). Now a program that will accurately do that alone would be interesting enough, but this program does much more.

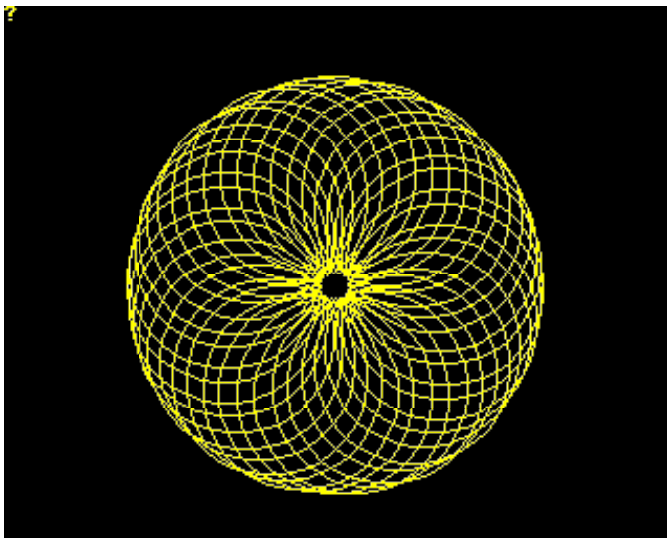


As an illustration imagine a six-sided figure, or hexagon. If we instruct the computer to draw only five sides of the six and then break away and start drawing another hexagon, it takes but little imagination to see that if the same rules are followed, a complex pattern will be produced that eventually assumes a symmetry and winds up where it started from. In the program, this is called a BLOCK.

Now suppose that upon completion of a block, the program breaks away yet again to draw another block - then another and another - and the most complex and beautiful patterns can emerge before once again the drawing point winds up where it started.

This is what the program does. In order to execute the design, the program needs to know (a) the number of sides of the base polygon, (b) the number of those sides it must draw before breaking away, (c) the regular length of side, (d) the angle it must break away after drawing the part-polygon, and (e) the angle that it must break away after completing a block. In practice, (d) and (e) are best identical, so I have left them like that.

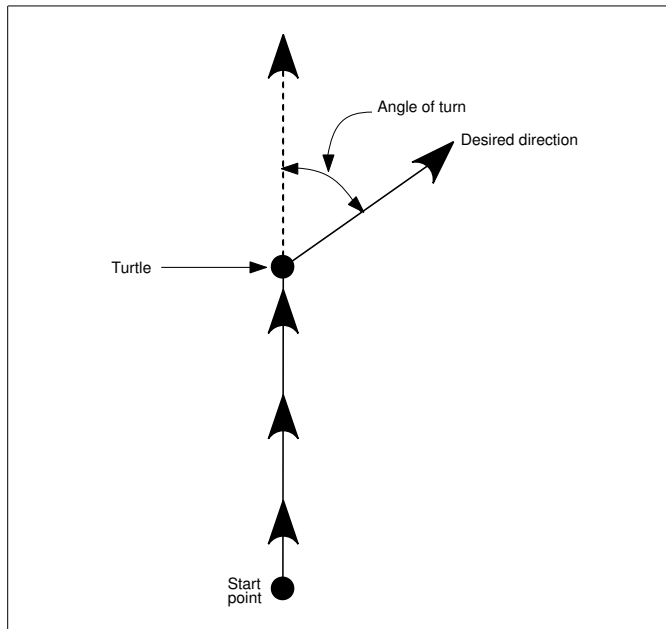
The turtle of my title? Well that comes about from LOGO and 'turtle graphics'. We may consider the drawing point as a turtle or some object capable of obeying instructions. At any time the turtle is not 'aware' of its position on the screen, but simply obeys two orders: go forward a certain distance, or turn through a certain angle.



One remarkable thing about this program is that although the drawing point is calculated in this way - perhaps thousands of times - it executes the patterns without fault, precisely drawing over another line or through a particular point quite meticulously.

Unravelling the listing, the single polygon-drawing part is in lines 170 to 210, with the calculations for the sides taking place on lines 180 and 190. Line 200 is the only instruction-to-draw in the entire program. It will be seen that around that loop is another, with TURN counting the times through the loop. Hence a certain number of TURNS go to make up one BLOCK. Line 220 of course is the break-away calculation.

The program allows for either one block to be drawn (SINGLE) or several, equal to the number of sides on the basic polygon (MULTIPLE), so line 260 determines if we must stop or not after completion of a block. Line 140 is interesting. Consider our turtle again: it has come to the end of drawing a line and must now determine through how many radians it must turn before starting on its next line (see the diagram). Since this angle will be constant



throughout the drawing process it need be calculated only once, on line 140.

When trying the program for the first time, enter responses as follows:

```
SIDES? 8  
DRAWN? 7  
ADD? 3  
SIZE? (DEFAULT=100)? (RETURN)  
MULTIPLE OR SINGLE* (M-S)? (RETURN)
```

The screen will then clear and the pattern will be drawn immediately. It will be noticed that SIZE and SINGLE are default values, which means that you may press the RETURN key without typing anything. The ADD input is the number of angle increments to be added on break-away and of course a different value will produce a different pattern.

Sometimes the completed pattern may be either too big to fit onto the screen or else too small to distinguish and appreciate. This is where the SIZE input comes in, for by entering values greater or smaller than 100 one may expand or shrink the design.

When the design is completed, a question mark appears in the upper left corner - so as not to spoil the design - and one may appreciate the display for as long as one wishes. Pressing RETURN clears the screen ready for another design. If you have a printer capable of reproducing graphics, a screen dump routine would be marvellous for this program, and you could fill your study wall with beautiful computer-generated designs. Alternatively, tracing paper may be Sellotaped over the screen and the designs copied carefully. I suggest that one marks only the corners of patterns, joining up later with a good ruler and Indian ink. A pair of parallel rulers would be useful too. Children will love colouring the designs with crayon or paint, and it is a perfect activity when it is too cold or wet for them to play outside.

If you wish, you could alter the break-away angle after each block. Change line 240 to  $B=B+J*2*PI/S$ :MOVEX,Y and have another input for J at a new line 65.

```
10 REM - Tuttle
20 REM - A pattern-drawing screen turtle
30 MODE7:ON ERROR GOTO270
40 INPUT""SIDES",S
50 INPUT"DRAWN",D
60 INPUT"ADD",I
70 INPUT"SIZE? (DEFAULT=100)",G$
80 IF G$="" SIZE=100 ELSE SIZE=VAL(G$)
85 REPEAT
90 INPUT"MULTIPLE OR SINGLE* (M-S)",G$
100 UNTIL G$="M" OR G$="S" OR G$=""
110 IF G$="" G$="S"
130 MODEL:X=620:Y=510:GCOLOR,2
140 B=I*2*PI/S:MOVE X,Y
150 BLOCK=0
160 FOR TURN=1 TO S
170 FOR SIDE=1 TO D
180 A=SIDE*2*PI/S:A=A+B
190 X1=SIZE*COSA:Y1=SIZE*SINA
200 PLOT1,X1,Y1
210 NEXT SIDE
220 B=B+I*2*PI/S
230 NEXT TURN
240 B=B+I*2*PI/S:MOVE X,Y
250 BLOCK=BLOCK+1
260 IF G$<>"S" AND BLOCK<=S GOTO160
270 INPUT G$:GOTO20
```