SUM SORCERY

We continue our series of programming suggestions with a look at an age-old mathematical brain-teaser — the 'magic square'. This puzzle is an ideal candidate for experimentation: the solution is fairly straightforward, but its implementation on home computers involves us in solving some interesting problems.

A magic square is simply a grid of cells into which the positive integer numbers (1, 2, 3, 4, 5, etc.) are entered (zero is not allowed and no number should be used twice). The object of the exercise is to distribute these numbers in such a way that each column and each row will give the same total when the numbers are added together. The simplest magic square is a three-by-three box, and one possible solution is:

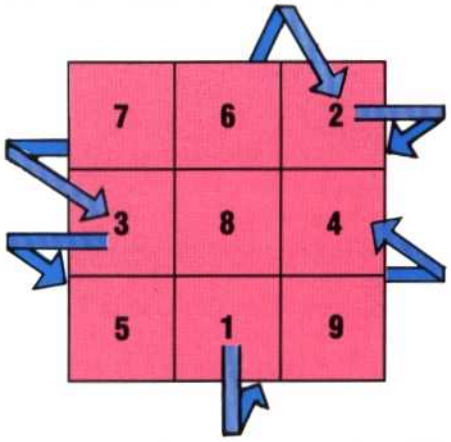
7	6	2	15
3	8	4	15
5	1	9	15
15	15	15	

In this example, all the rows and columns add up to 15. As an exercise, try the puzzle yourself — but this time use a five-by-five grid and enter the numbers from 1 to 25. You will find that this isn't as easy as it looks, and undoubtedly considerable trial and error will be needed before the correct solution is obtained.

Fortunately, your computer can make things easier. One answer is to write a program to fill the squares with numbers and then test each row and column individually. However, this method could well take many hours to find a solution for even quite small squares, seven or nine cells wide. What is needed is a method for generating the numbers. To save you the trouble of discovering your own, here's one that works every time:

1) Start with the number 1 in the middle cell on the bottom row.
2) After each cell is filled, move one cell down and one cell right before entering the next number. If moving right takes you off the right-hand edge of the grid, go to the first column instead. Similarly, if moving down takes you off the bottom of the grid then move to the top row. This is similar to a screen 'wraparound' effect.
3) If the next cell is already occupied, move one position to the left of the last cell used.
4) Carry on until you reach the last empty cell.

Applying this method to our example shows how a three-by-three square is generated:



It's very easy to write a program to do this for any size of square. Simply create an empty two-dimensional array of the correct size for the desired magic square, and then use a loop to fill it in according to the rules given. You should note that the method works only for magic squares with

Squares of Enchantment

These squares were generated by the program and have been successfully checked by it — as the printed message shows. In the 15 × 15 square, a diagonal pattern of three- and two-digit numbers can be observed, the consequence of the construction algorithm

52	42	32	22	12	2	73	72	62
63	53	43	33	23	13	3	74	64
65	55	54	44	34	24	14	4	75
76	66	56	46	45	35	25	15	5
6	77	67	57	47	37	36	26	16
17	7	78	68	58	48	38	28	27
19	18	8	79	69	59	49	39	29
30	20	10	9	80	70	60	50	40
41	31	21	11	1	81	71	61	51

IAN McKINNELL