an odd number of cells — your program should reject even-numbered grids.

Care should be taken when the square is displayed. In the interest of neatness, all numbers should be aligned correctly in rows and columns. This is achieved quite simply if your micro features the PRINT USING command. If not, it is best to convert the number to be printed into a string. This can then be padded out with space characters so that the 'number' is always the same length. A subroutine to achieve this is:

```
1000 REM Convert A to AS and align
1010 AS=STRS(A)
1020 IF LEN(AS) <3 THEN AS=" "+AS:GOTO 1020
1030 RETURN
```

The exact method, of course, will depend on the particular computer being used.

The next problem is the screen size — most micros will be unable to display large magic squares on-screen. A 40-column screen has room for 13 two-digit columns, but a 13-by-13 square will include some three-digit numbers, so a nine-by-nine square is the largest that may be displayed. A printer will allow much larger squares to be generated. Most printers have a maximum width of 80 or 132 columns, and larger squares may be printed in sections that are joined together later.

The overall aim of this project is to create the largest magic square you can, and present it as neatly as possible. As an experiment, try writing a trial and error program and determine how long it takes to find an answer (although bear in mind that this will take a ridiculously long time to achieve a result). Alternatively, you could look for even faster methods for generating the squares.

130	114	98	82	66	50	34	18	2	211	210	194	178	162	146
147	131	115	99	83	67	51	35	19	3	212	196	195	179	163
164	148	132	116	100	84	68	52	36	20	4	213	197	181	180
166	165	149	133	117	101	85	69	53	37	21	5	214	198	182
183	167	151	150	134	118	102	86	70	54	38	22	6	215	199
200	184	168	152	136	135	119	103	87	71	55	39	23	7	216
217	201	185	169	153	137	121	120	104	88	72	56	40	24	8
9	218	202	186	170	154	138	122	106	105	89	73	57	41	25
26	10	219	203	187	171	155	139	123	107	91	90	74	58	42
43	27	11	220	204	188	172	156	140	124	108	92	76	75	59
60	44	28	12	221	205	189	173	157	141	125	109	93	77	61
62	46	45	29	13	222	206	190	174	158	142	126	110	94	78
79	63	47	31	30	14	223	207	191	175	159	143	127	111	95
96	80	64	48	32	16	15	224	208	192	176	160	144	128	112
113	97	81	65	49	33	17	1	225	209	193	177	161	145	129

```
10 REM************************
15 REM***MAGIC SQUARES********
20 REM******SET-UP*************
30 M=19:DIM A(M,M)
40 PRINT:PRINT"Magic Squares"
50 PRINT:PRINT"How many rows (1 to 19)";
:INPUT S
60 IF S<0 OR S<>INT(S) THEN PRINT"ERROR":GOTO
50
70 IF S>M THEN PRINT"ERROR":GOTO 50
80 IF S/2=INT(S/2) THEN PRINT"ERROR - Odd
Numbers Only":GOTO 50
90 REM**GENERATE SQUARE********
100 X=INT(S/2)+1:Y=S:C=1
110 A(X,Y)=C
120 C=C+1:IF C>S*S THEN GOTO 200
130 X=X+1:IF X>S THEN X=1
140 Y=Y+1:IF Y>S THEN Y=1
150 IF A(X,Y)<>0 THEN X=X-2:Y=Y-1
160 IF Y=0 THEN Y=S
170 IF X=0 THEN X=S
180 IF X=-1 THEN X=S-1
190 GOTO 110
200 REM***PRINT SQUARE*********
210 PRINT:PRINT
220 FOR Y=1 TO S:FOR X=1 TO S
230 A=A(X,Y):GOSUB 380:PRINT" ";A$;" ";
240 NEXT X:PRINT:NEXT Y

250 REM***CHECK ROWS & COLS****
260 F=0
270 FOR Y=1 TO S:T=0
280 FOR X=1 TO S:T=T+A(X,Y):NEXT X
290 IF F=0 THEN U=T:F=1
300 IF T<>U THEN PRINT"ERROR - Row 1 &
Row";Y;" Do Not Match":STOP
310 U=T:NEXT Y
320 FOR X=1 TO S:T=0
330 FOR Y=1 TO S:T=T+A(X,Y):NEXT Y
340 IF T<>U THEN PRINT"ERROR - Row 1 &
Col";X;" Do Not Match":STOP
350 U=T:NEXT X
360 PRINT:PRINT"All rows and cols add to ";T
370 STOP
380 REM*****NUM-STRING CONV*****
390 A$=STR$(A)
400 IF LEN(A$)<3 THEN A$=" "+A$:GOTO 400
410 RETURN
```

Basic Flavours

This program is written in Microsoft BASIC, so it should run unchanged on most popular micros. Spectrum owners must insert LET before all assignment statements. The program asks for the number of rows (and, therefore, columns) in the Magic Square, and checks that this is a positive, integral, odd number. It computes and displays the Magic Square, and then, from line 250 on, checks its own output. If this seems unnecessary, then omit lines 250-360.