Assignment Introduction . Magic Squares and Associative Magic Squares

A **magic square** of order n is a square in which all n^2 integers are distinct, greater than 0 and are chosen in such a way, that the n numbers in each row, column and diagonal of the square add up to the same constant. This constant is called the **magic constant**.

A normal or pure square is restricted to containing the integers from 1 to n^2. In the remainder of this text we shall only consider normal squares. In figure 1 you can see a magic square of order 4, with magic constant 34.

9	6	3	16
4	15	10	5
14	1	8	11
7	12	13	2

Figure 1: Magic Square with Magic Constant 34

An **associative magic square** is a special case of a magic square. It fulfills all the requirements of a magic square, with the additional property that every pair of numbers symmetrically opposite to the center, sum up to $n^2 + 1$, where n is the order of the square. As an example, take a look at the matrix portrayed in figure 2.

A	B	C	D
E	F	G	H
I	J	K	L
M	N	O	P

Figure 2: Square of order 4

In this square the following pairs should sum up to $4^2 + 1 = 17$: (A-P), (B-O), (C-N), (D-M), (E-L), (F-K), (G-J) and (H-I). If the order of the square is odd, the center entry should be taken twice to form a pair with itself. Figure 3 helps to illustrate this. Besides the pairs (A-Y), (BX) etc, the pair (M-M) should sum up to 26.

A	B	C	D	E
F	G	H	I	J
K	L	M	N	O
P	Q	R	S	T
U	V	W	X	Y

Figure 3 - Square of Order 5

An example of an associative magic square with magic constant 34 is shown in figure 4.

2	13	11	8
16	3	5	10
7	12	14	1
9	6	4	15

Figure 4 - Associative Magic Square with Magic Constant 34

Assignment
You are asked to write a program that is able to check whether a given square is a magic square, an associative magic square (and thus also a magic square) or neither. Your program must be capable of checking squares of any order!
The input should be a matrix (2-dimensional array) of integers. You can choose to provide the input by letting your program read the numbers from a file or you can choose to include the matrix .hardcoded. in your source code.

The output should be in the form of some print statements to the Dos Prompt. First, print if the matrix is a normal square. Subsequently, print out the sum of every row, column, diagonal and pairs symmetrically opposite to the center and the answer whether the given matrix is a magic square, associative magic square or neither. If it is not magic or associative magic, you should print why not (e.g. .The pairs (3, 7) and (5, 4) add up to a different total.).

Your program should at least contain the following components:

2 A method called *isNormal* to check if all the numbers in the matrix correspond to the form 1… n^2(i.e. check if the matrix is a normal square).

2 Methods *checkRows, checkColumns and checkDiagonals* to check if the rows, columns and diagonals, respectively, add up to the same magic constant.
3 A method called *checkPairs* to check if all pairs symmetrically opposite to the center sum up to $n^2 + 1$.

You can use the magic square in figure 1 and the associative magic square in figure 4 to test your
program. For your convenience, some more magic and associative magic squares are depicted
below. Use them to check if your program works properly.

NOTE:
This is NOT an exercise about inheritance, polymorphism or interfaces! You should, therefore, refrain from extending a superclass, creating an abstract class, implementing an interface etc.

3	7	14	16	25
11	20	23	2	9
22	4	6	15	18
10	13	17	24	1
19	21	5	8	12

Figure 5 - Magic Square with Magic Constant 65

51	50	3	2	63	62	1	14
52	4	49	64	1	16	61	13
5	53	48	33	32	17	12	60
6	47	54	31	43	11	18	59
46	7	30	55	10	35	58	19
45	29	8	9	56	57	36	20
28	44	43	42	23	22	21	37
27	26	25	24	41	40	39	38

Figure 6 - Magic Square with Magic Constant 260