## Exercise 15: MD Array

1. Write one or more statements to perform each operation.
   1. Declare an array containing three rows (numbered one to three) and five columns (numbered one to five) of integers.

*var sample : array 1 .. 3, 1 .. 5 of int*

* 1. Set all the elements of the array to zero.

*for initializeCurrentRow : 1 .. 3 by 1*

*for initializeCurrentColumn by 1*

*sample (initializeCurrentRow, initializeCurrentColumn) := 0*

*end for*

*end for*

* 1. Find the sum of all the elements in the array and store this sum in grandTotal.

*var grandTotal : int := 0*

*for addCurrentRow : 1 .. 3 by 1*

*for addCurrentColumn by 1*

*grandTotal += sample (addCurrentRow, addCurrentColumn)*

*end for*

*end for*

* 1. Find the sum of the elements in the second row and store this value in row2sum.
  2. Find the sum of the elements in the third column and store this value in col3sum.

*var col3sum : int := 0*

*for addCurrentRow1 : 1 .. 3 by 1*

*col3sum := sample (addCurrentRow1, 3)*

*end for*

* 1. Find the sum of all negative elements in the array and store this value in negSum.

*var negSum : int := 0*

*for addCurrentRow2 : 1 .. 3 by 1*

*for addCurrentColumn2 by 1*

*if sample (addCurrentRow2, addCurrentColumn2) = 0 then*

*negSum := sample (addCurrentRow2, addCurrentColumn2)*

*end if*

*end for*

*end for*

* 1. Replace each element by its square.

*for addCurrentRow : 1 .. 3 by 1*

*for addCurrentColumn by 1*

*sample (addCurrentRow, addCurrentColumn) := sample (addCurrentRow, addCurrentColumn) \*\* 2*

*end for*

*end for*

* 1. Find the largest value in the array and store this value in largest.

1. A magic square is a square array of numbers in which the sums of the rows, the columns, and the diagonals are all equal. Write a program that prompts the user for the dimension of the square, reads integers into the array accordingly, print out your square and determines whether or not they form a magic square.

For example, the following is a magic square.

