Magic Square

Sample AP Free Response Question

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

An *n x n magic square* is a square array of *n2* distinct integers arranged such that the n numbers along any row, column, major diagonal, or minor diagonal have the same sum. Shown below are two magic squares

Each row, column, and diagonal add up to 34

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

Each row, column, and diagonal add up to 15

| 8 | 1 | 6 |
| --- | --- | --- |
| 3 | 5 | 7 |
| 4 | 9 | 2 |

The MagicSquare class has one private instance variable, a two-dimensional array mySquare.

Download MagicSquare and MagicSquareTester. The constructor is created. You will need to implement the following methods.

public int sumRow(int row)

public int sumCol(int col)

public int sumMajorDiag()

public int sumMinorDiag()

public boolean isMagic()

isMagic() should call the other methods to determine if the square is magic or not.

The **major diagonal** starts in the upper-left corner and proceeds to the bottom-right corner *(8, 5, 2 in the second example).*

The **minor diagonal** starts in the bottom-left corner and proceeds to the upper-right corner *(4, 5, 6 in the second example).*

Be sure to run MagicSquareTester to test your code!