Worksheet W.A.21.1 Name _____

# Magic Square

Background:

A magic square is an *N* x *N* square of numbers with these characteristics:

1. Every number from 1 through $N^2$ must appear just once.
2. Every row, column, and diagonal must add up to the same total.

The following is an example of a 4 x 4 magic square:

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

To solve this programming problem, several routines should be developed. This worksheet will ask you to write solutions to the following algorithms.

```
int sumRow (int[][] square, int row);
//  Precondition:  square is an initialized matrix, MAX rows x MAX columns
//                 0 <= row < MAX
//  Postcondition: returns the sum of the values in row

boolean unique(int[][] square)
//  Precondition: square is initialized with integers.
//  Action: Inspects every value in square, checking that each one is
//          a unique integer ranging from 1..MAX*MAX
//  Postcondition: Returns true if each value is unique from 1..MAX*MAX,
//                 otherwise returns false
```

**Add methods to sumCol and SumDiag.**

**public boolean** testMagic(**int**[][] square)

```
//  Precondition: square is initialized with integers.
//  Action: Checks that row, col, and diagonal sums are equal and elements are unique
//  Postcondition: Returns true if magic else return false.
//
```

```
//Test your methods with the following arrays.
public static void main(String[] args)
             int[][]  one ={{16, 3, 2,13},
                {5, 10, 11,8},
                {9, 6, 7,12},
                {4, 15, 14,1}};

   int[][] two ={{14, 5, 2,4},
                {3, 12, 6,8},
                {9, 11, 7,10},
                {13, 15, 16,1}};

   int[][] three ={{14, 5, 2},
                {3, 12, 6},
                {9, 11, 7}};
   int[][] four ={{32,29, 4,1,24,21},
                     {30,31, 2,3,22,23},
                     {12,9, 17,20,28,25},
                     {10,11, 18,19,26,27},
                     {13,16, 36,33,5,8},
                     {14,15, 34,35,6,7}};

int[][] five ={{1, 2, 3,4},
             {3, 4, 1,2},
             {4, 1, 2,3},
             {2, 3, 4,1}};

  int[][] six ={{1, 1, 1},
                {1, 1, 1},
                {1, 1, 1}};

MagicSquare tms  = new MagicSquare();
 tms.printTable(one);
 System.out.println("MAGIC SQUARE? " + tms.testMagic(one));
 System.out.println();
 tms.printTable(two);
 System.out.println("MAGIC SQUARE? " +  tms.testMagic(two));
 System.out.println();
 tms.printTable(three);
 System.out.println("MAGIC SQUARE? " +  tms.testMagic(three));
 System.out.println();
 tms.printTable(four);
 System.out.println("MAGIC SQUARE? " + tms.testMagic(four));
 System.out.println();
 tms.printTable(five);
 System.out.println("MAGIC SQUARE? " +  tms.testMagic(five));
 System.out.println();
 tms.printTable(six);
 System.out.println("MAGIC SQUARE? " + tms.testMagic(six));
}
```