# MagicSquares

# Adding onto your Iterations, create a new class called MagicSquares.

**MagicSquares Method 1. Perfect Square problem**

**public static boolean perfectSquare(int num)**

Some perfect squares have unique mathematical properties.

For example, 36 is a perfect square, 62

Write a method that **returns** if a number is a perfect square.

**MagicSquares Method 2. Sum of Sequence problem**

**public static boolean sumOfSeq(int num)**

Some perfect squares have unique mathematical properties.

For example, 10 is a sum of the numbers (1+2+3+4 = 10)

Write a method that **returns** if a number is equal to the sum of consecutive integers beginning with 1.

**MagicSquares Method 3. Magic Square problem**

**public static void magicSquare(int n)**

A “magic square” (any number that is both a perfect square AND equal to the sum of consecutive integers beginning with 1)

Using your perfectSquare method and sumOfSeq method, write a method that prints the first **n** magic squares.

**Inside your Main Method in IterTester**

Find the first four magic squares. The first one is the integer 1.