```
package Library;

public class MagicSquare {

		private int [][] theSq;

		public MagicSquare (int [][] sqData) {
			theSq = sqData;
		}

		private int colSum (int c) {
			int sum = 0;
			for (int i = 0; i < theSq.length; i++) {
				sum += theSq[i][c];
			}
			return sum;
		}

		private int rowSum (int r) {
			int sum = 0;
			for (int i = 0; i < theSq[r].length; i++) {
				sum += theSq[r][i];
			}
			return sum;
		}

		private int diag1 () {
			int sum = 0;
			for (int i = 0; i < theSq.length; i++) {
				sum += theSq[i][i];
			}
			return sum;
		}

		private int diag2 () {
			int sum = 0;
			for (int i = theSq.length-1; i >= 0; i--) {
				sum += theSq[i][i];
			}
			return sum;
		}

		private Boolean valuesCheck () {
			int sum = 0;
```

```
        for (int i = 0; i < theSq.length; i++) {
                for (int j = 0; j < theSq[i].length; j++) {
                        sum += theSq[i][j];
                }
        }

        int a = theSq.length*theSq.length;

        if (sum != (a)*(a+1)/2) {
                return false;
        }
        return true;
}

public Boolean isMagic () {
        int B = rowSum(0);
        if (B != this.diag1 () || B != this.diag2 () || this.valuesCheck() != true) {
                return false;
        }

        for (int i = 0; i < theSq.length; i++) {
                if (B != rowSum(i) || B != colSum(i)) {
                        return false;
                }
        }

        return true;
}

private int cornerSum () {
        int sum = 0;
        int a = theSq.length - 1;
        sum += theSq[0][0] + theSq[a][a] + theSq[0][a] + theSq[a][0];
        return sum;
}

private int centerSum () {
        int sum = 0;
        double k = (theSq.length-1)/2.0;
        int a1 = (int) Math.floor(k);
        int a2 = (int) Math.ceil(k);

        if (a1 != a2) {
                sum += theSq[a1][a2] + theSq[a2][a1] + theSq[a1][a1] + theSq[a2][a2];
```

```
        }

        else {
                sum = theSq[a1][a1];
        }

        return sum;

}

private int ULSum () {
        return theSq[0][0] + theSq[0][1] + theSq[1][0] + theSq[1][1];
}

private int URSum () {
        int n = theSq.length-1;
        return theSq[0][n] + theSq[0][n-1] + theSq[1][n] + theSq[1][n-1];
}

private int LLSum () {
        int n = theSq.length-1;
        return theSq[n][0] + theSq[n][1] + theSq[n-1][0] + theSq[n-1][1];
}

private int LRSum () {
        int n = theSq.length-1;
        return theSq[n][n] + theSq[n][n-1] + theSq[n-1][n] + theSq[n-1][n-1];
}

private int topBottomCenter()  {
        int sum = 0;
        double k = (theSq.length-1)/2.0;
        int n = theSq.length-1;
        int a1 = (int) Math.floor(k);
        int a2 = (int) Math.ceil(k);

        if (a1 != a2) {
                sum += theSq[n][a2] + theSq[0][a1] + theSq[n][a1] + theSq[0][a2];
        }

        else {
                sum = theSq[0][a1] + theSq[n][a1];
        }
```

```
        return sum;
}

private int leftRightCenter() {
        int sum = 0;
        double k = (theSq.length-1)/2.0;
        int n = theSq.length-1;
        int a1 = (int) Math.floor(k);
        int a2 = (int) Math.ceil(k);

        if (a1 != a2) {
                sum += theSq[a1][0] + theSq[a2][0] + theSq[a1][n] + theSq[a2][n];
        }

        else {
                sum = theSq[a1][0] + theSq[a1][n];
        }

        return sum;
}


public boolean isA4x4Durer() {
        for (int i = 0; i < theSq.length; i++) {
                if (theSq[i].length != theSq.length) {
                        return false;
                }
        }

        int B = rowSum(0);

        if (this.isMagic() == false || this.cornerSum() != B ||
                        this.centerSum() != B || this.ULSum() != B ||
                        this.URSum() != B || this.LLSum() != B ||this.LRSum() != B ||
                        this.topBottomCenter() != B ||   this.leftRightCenter() != B)
        {
                return false;
        }

        return true;


}
```

```
	public String toString () {
		String ret = "";

		for (int r = 0; r < theSq.length; r++) {
			for (int c = 0; c < theSq[r].length; c++) {
				ret += theSq[r][c] + " ";
			}
			ret += "\n";
		}

		return ret;
	}


}
```

TESTER

```
package Library;

public class MagicSquareTester {

	public static void main(String[] args) {
		// TODO Auto-generated method stub

		int[][] anySq1 = { {2,7,6}, {9,5,1}, {4,3,8} }; //MAGIC, NOT DURER
		int [][] anySq2 = { {16,3,2,13},{5,10,11,8},{9,6,7,12},{4,15,14,1} }; //MAGIC, DURER
		int [][] notMagic1 = { {1,1,1},{1,1,1},{1,1,1} }; //NOT MAGIC
		int[][] notMagic2 ={ {1,1,1,1},{1,1,1,1},{1,1,1,1},{1,1,1,1} };  //NOT MAGIC
		MagicSquare ms = new MagicSquare(anySq1);  //testing a 3x3 magic square
		MagicSquare ms2 = new MagicSquare(anySq2); //testing a 4X4 durer magic square
		MagicSquare ms3 = new MagicSquare(notMagic1);
		MagicSquare ms4 = new MagicSquare(notMagic2);


		//System.out.print(ms2.centerSum());



		if (ms.isMagic())
			System.out.println (ms + " is magic!");
		else
			System.out.println(ms + " is NOT magic!");
```

```
            if (ms.isA4x4Durer())
                  System.out.println(ms + " is a Durer magic!");
            else
                  System.out.println(ms + " is NOT a Durer Magic Square");

            if (ms2.isMagic())
                  System.out.println (ms2 + " is magic!");
            else
                  System.out.println(ms2 + " is NOT magic!");

            if (ms2.isA4x4Durer())
                  System.out.println (ms2 + " is a Durer magic!");
            else
                  System.out.println(ms2 + " is NOT a Durer Magic Square");

            if (ms3.isMagic())
                  System.out.println (ms3 + " is magic!");
            else
                  System.out.println(ms3 + " is NOT magic!");

            if (ms3.isA4x4Durer())
                  System.out.println (ms3 + " is a Durer magic!");
            else
                  System.out.println(ms3 + " is NOT a Durer Magic Square");

            if (ms4.isMagic())
                  System.out.println (ms4 + " is magic!");
            else
                  System.out.println(ms4 + " is NOT magic!");

            if (ms4.isA4x4Durer())
                  System.out.println (ms4 + " is a Durer magic!");
            else
                  System.out.println(ms4 + " is NOT a Durer Magic Square");

      }

}
```