Problem 4: Write a Java method ***public static void magicOdd(int[][] square)*** that creates a magic square of an ***odd*** size using the following algorithm:

1. The number ***1*** goes in the middle of the top row;
2. All numbers are then placed one ***column to the right*** and ***one row up*** from the previous number;
3. Whenever the next number placement is above the top row, stay in the same column and place the number in the bottom row (note the place of ***2*** instead of the shaded location);
4. Whenever the next number placement is outside of the rightmost column, stay in the same row and place the number in the leftmost column (note the place of ***3*** instead of the shaded location);
5. When encountering an already filled-in square, place the next number directly below the previous number;
6. When the next number position is outside both a row and a column, place the number directly beneath the previous number (note the place of ***7*** instead of the shaded location).

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

```
Public static double[] sort (double[] arr){
    double temp;
        for (int q = arr.length ; q>0; q --)
        for ( int i =0; i < q-1; i++) {
                if (arr [i] > arr [i+1]) {
                    temp = arr [i];
                    arr [i+1] = arr[i];
                    arr [i+1] = temp;

                }

        }
    return arr;
```

2

OOP.MT.170317.H008