AP CS Recursion Assignment Part 2

Recursive Programming

Create your own class called LastnameRecursion. The class needs to contain a main method with calls to multiplyEvens and writeSquares for each of the cases below and needs to contain the following two methods. The result from the multiplyEvens needs to be printed from the main method. The result from the writeSquares method is printed within the writeSquares method.

1. Write a method `multiplyEvens` that returns the product of the first n even integers. For example:

`multiplyEvens(1)` returns 2

`multiplyEvens(2)` returns 8 (2 * 4)

`multiplyEvens(3)` returns 48 (2 * 4 * 6)

`multiplyEvens(4)` returns 384 (2 * 4 * 6 * 8)

Your method should throw an IllegalArgumentException if passed a value less than or equal to 0. You may NOT use a while loop, for loop or do/while loop to solve this problem; you MUST use recursion.

2. Write a method `writeSquares` that takes an integer n as a parameter and that writes the first n squares to System.out separated by commas with the odd squares in descending order followed by the even squares in ascending order. For example, the call:

```
writeSquares(5);
```

should produce the following output:

```
25, 9, 1, 4, 16
```

The odd squares (25, 9, and 1) appear first in descending order followed by the even squares (4 and 16) in ascending order. Notice that commas are used to separate consecutive values in the list. Your method should send its output to System.out and should not call println. For example, the following calls:

```
writeSquares(5);
System.out.println();  // to complete the line of output
writeSquares(1);
System.out.println();  // to complete the line of output
writeSquares(8);
System.out.println();  // to complete the line of output
```

should produce exactly three lines of output:

```
25, 9, 1, 4, 16
1
49, 25, 9, 1, 4, 16, 36, 64
```

You must exactly reproduce the format of these examples. Your method should throw an IllegalArgumentException if passed a value less than 1. You may NOT use a while loop, for loop or do/while loop to solve this problem; you must use recursion.