CS-1181 Lab Problem 9: (Matching)

PURPOSE: To review and practice writing recursion

DIRECTIONS:

Part A (Due by end of first lab session)

Write a recursive method called `countDown(start, stop)` that counts backwards from start to stop (inclusive). If stop >= start, your method should simply exit (see example output below). Your main program should test your method using the following three calls:

```
countDown(10, 3);
countDown(4, 5);
countDown(-2, -6);
```

EXAMPLE:

```
Counting down from 10 to 3:
10 9 8 7 6 5 4 3

Counting down from 4 to 5:

Counting down from -2 to -6:
-2 -3 -4 -5 -6
```

Part B

Create a class called `Matching` with a recursive method called `nestParen` with the following signature:

```
public static boolean nestParen (String n)
```

Your method should return if **n** is a nesting of zero or more pairs of parenthesis, like “(())” or “((()))” and otherwise. Please see example return values for any questions regarding test cases.

EXAMPLE:

- `nestParen("(())")` → `true`
- `nestParen("((()))")` → `true`
- `nestParen("(((x))")` → `false`
- `nestParen("((())")` → `false`
- `nestParen("((()()")` → `false`
- `nestParen("")` → `true`
- `nestParen("(yy)")` → `false`
- `nestParen("((yy()))")` → `false`

For full credit, ensure that your program is well commented and follows JavaDoc standards for your method(s). Comments are only required for the Part B segment of the lab.

Note: This problem is based on one from codingbat.com. There are many other recursion problems there if you desire more practice.

RUBRIC:

[1pt]	**Documentation**
[1pt]	**Part A correct**
[1pt]	**Part B correct**