

CS 1181 - Computer Science II

Practice Problem: (Matching)

Purpose: To review and practice writing recursion.

Part A:

Write a recursive method called `countDown(start, stop)` that counts backwards from `start` to `stop` (inclusive). If `stop >= start`, your method should print the input parameters and 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);
```

Part A example output:

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 `true` if `n` is a nesting of zero or more pairs of parenthesis, like `"()"` or `"((()))"` and `false` otherwise. Please see example return values for any questions regarding test cases.

Part B example return values:

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

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

Rubric:

[/1] Documentation

[/1] Part A correct

[/1] Part B