Lab 6: Recursion Practice

DUE: Friday, 17 Mar, 5:00pm

Preparation

Nothing to turn in for Wednesday, but you should think about how to solve all of these problems for Wednesday. Lab on Wednesday will meet in TCL312 where we will use the FreeBSD machines for this lab.

Lab Description

For this lab, you are to implement a series of recursive functions. The functions themselves will produce the output, so we will not have the main method handling all I/O for this lab. This lab will have one Class called `Recur.java` with a main method and a bunch of recursive functions and no constructor. Your class should implement the `Recurface` interface found in the shared cs136 directory (and printed on the other side of this handout).

1. Write a recursive method that is passed a value n and returns the summation of 1 to n.
2. Write a recursive method that is passed a value n and prints to the screen a countdown from n to 1 on separate lines and then "Liftoff!" on the final line.
3. Write a recursive method that is passed a value n and returns the factorial $n!$
4. Write a recursive method that is passed a value n and prints to the screen a count UP from 1 to n on separate lines.
5. Write a recursive method that is passed a value n and returns the value of the nth number in the Fibonnaci sequence (the Fibannaci sequence starts with 1 1 and each subsequence number in the sequence is the sum of the two previous numbers in the sequence).
6. Write a recursive method that is passed a value n and prints to the screen a countdown from n to 2 followed by the word "stop" and then a count up from 2 to n all on separate lines.
7. Write a recursive function to print out the characters of a string with spaces between the characters. Make sure your methods does not print out a leading or trailing space, unless it is a leading or trailing character of the original string.

To test your functions, write a `main` method that is passed an integer on the command line. It should first print to the screen your name after being run through the spacing function (#7 above). It should then call each of the six recursive methods with the command line argument value as n. You will find the function:

```
int Integer.parseInt(String str)
```

helpful. It takes a string representing an integer as an argument, and returns the integer value of the string. If the user tries to run your program with no arguments, they should be given instructions as to how the program should be called.

Submitting Your Work

When you're finished, create and submit a gzipped tar file "youruserid_lab6.tar" of the directory with the code in it.

Recurface Interface

```
public interface Recurface {

    // #1: sums integers from 1 to n
    public int sum(int n);

    // #2: counts down from n to 1 then prints Liftoff!
    public void countDOWN(int n);

    // #3: computes n factorial
    public int fact(int n);

    // #4: counts up from 1 to n
    public void countUP(int n);

    // #5: returns the nth Fibanacci number
    public int fib(int n);

    // #6: counts down from n to 2, prints stop, then counts back up to n
    public void downup(int n);

    // #7: prints out str with spaces between letters (no trailing/leading blank)
    public void spaceStr(String str);

}
```