CSE 1002: Fun with SquaresFun with Squares

*Due: September 27th, 2023, **8pm**September 27th, 2023, **8pm***

Additional practice is always available at CodingBat Java, Project Euler, and Kattis.

Background

Recursion

Recursion is the process a procedure goes through when one of the steps of the procedure involves invoking the procedure itself. A procedure that goes through recursion is said to be 'recursive'.

A method is said to exhibit a recursive behavior when it is exhibits a behavior that defined by the following properties

- A Simple Base case (or cases): Which is a terminating scenario that does not produce an answer
- A recursive step — a set of rules that reduces all successive cases toward the base case.

Types of Recursion

The Most common Types of Recursion Includes

- Direct Recursion: This is the type of recursion where the function calls itself from within the function
- Indirect Recursion: This is the type of recusion where one function calls another function and eventually resulting in the original function being called again

For the purposes of this lab we will be using Direct Recursion to draw squares using the StdDraw library used in the previous lab

Read about Recursion in Wikipedia or our textbook.

How To Use StdDraw

Download the file stddraw.jar and keep it in your working directory. Consult the documentation for additional details.

The following commands show how to use tje jar file when compiling and running your program.

Windows (semicolon before stddraw.jar):

```
    javac -cp .;stddraw.jar Squares.java
    java -cp .;stddraw.jar Squares 2.2 4 3
```

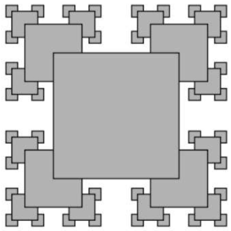
Mac/Linux/Unix (colon before stddraw.jar):

```
    javac -cp .:stddraw.jar Squares.java
    java -cp .:stddraw.jar Squares 2.2 4 3
```

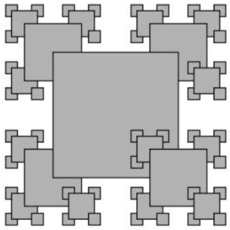
The stddraw.jar and your .java file should be in the same folder. Here is an example of a program using the stddraw methods: Here!

Task

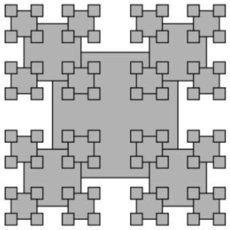
Write a program to draw one of four recusively defined patterns. These are the patterns:



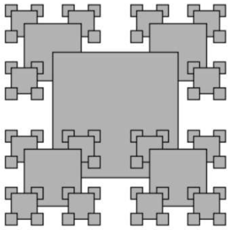
Pattern-1



Pattern-2



Pattern-3



Pattern-4

In each pattern a smaller square is drawn at each of the four corners of the next larger square. This pattern repeats again and again. It might repeat any number of times or levels. In the pictures above, the pattern is repeated four times or levels. If another level were asked for, then 192 very small squares would have to be drawn. The patterns differ in which squares are drawn on top of the others and thus may be hidden from view.

The largest square, the one at level one, is drawn in the center of a square canvas and the length of its side is one half the length of the side of the canvas. The center of the next smaller square is always at the corner of the larger square. And the size of the next smaller square is determined by a fixed ratio R. If the smaller square has a side of length s and the next larger square has a side of length b, then R is the ratio b/s. In the pictures above the ratio R is about 2.2.

Input

Read ratio R, level L and pattern number n (1<=n<=4) as command line arguments.

Output

Draw the specific square pattern using StdDraw.

Turning it in

Turn in your Java source code for the program using the submission server. The beginning of your source file must include a header like this:

```
/*
 * Author: name, e-mail address
 * Course: CSE 1002, Section 7, Fall 2023
 * Project: Fun with SquaresFun with Squares
*/
```

The file name to submit must be `Square.javaSquare.java` and the project id is `squaresquare`.

✓	Java file Choose File No file chosen Analyze the file (this may take a while): Analyze Reset!	Java	✓	Java file Choose File No file chosen Analyze the file (this may take a while): Analyze Reset!	
	File 1 Choose File No file chosen Add another file Remove last file Control code: Submit Reset!	file=`Square.java` **Course=cse1002** **Project=square**		File 1 Choose File No file chosen Add another file Remove last file Control code: Submit Reset!	fi **C** **I**

All submissions are checked for similarity against other programs. If the amount of similarity is unusually high, the submission will receive no credit regardless of the reason.

All submissions are checked for similarity against other programs. If the amount of similarity is unusually high, the submission will receive no credit regardless of the reason.