Recursion Design Patterns

The recursive designs we've seen share some common steps:

1. Some **action** to draw the fundamental part of the design (a box, a line of characters, etc.)
2. **Recursion** - a invocation of the same function to repeat the design.

And in the case of Turtle drawings, there's sometimes a…
3. **Setup** - the movement used to get the Turtle into position for the next recursion
4. An "**invariation**", the movement used to get the Turtle back into the position it started at.

Every recursive design we're creating can be broken down into some combination of these steps. Understanding this can be useful, because it gives you some scaffolding to work with when you approach a new problem.

The rest of this handout contains examples of recursive patterns you've seen.
For each pattern, the corresponding design pattern is shown.
Study this handout with the comparison code, **recursive-design-patterns.py**

Design Questions

What's the base case?

What's the fundamental action?
(Print a line, draw a square, etc.)

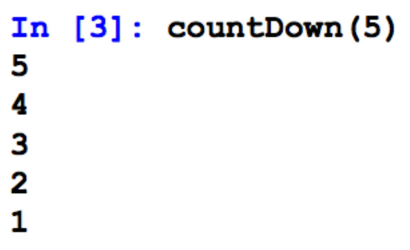
How many times will it recurse?

Is setup needed? (Turtles only)

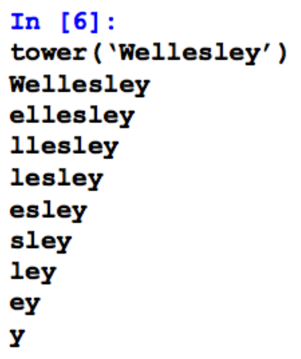
Does it "invariate"? (Turtles only)

Character Patterns

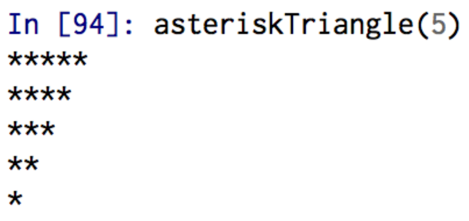
act
recurse



```
In [3]: countDown(5)
5
4
3
2
1
```

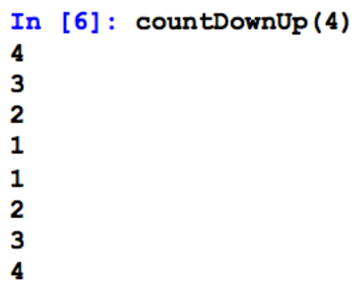


```
In [6]:
tower('Wellesley')
Wellesley
ellesley
llesley
lesley
esley
sley
ley
ey
y
```

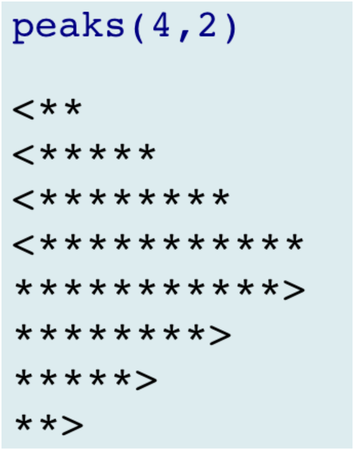


```
In [94]: asteriskTriangle(5)
*****
****
***
**
*
```

act
recurse
act



```
In [6]: countDownUp(4)
4
3
2
1
1
2
3
4
```



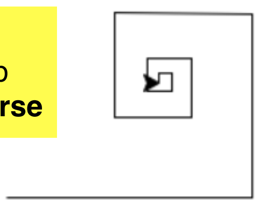
```
peaks(4,2)

<**
<*****
<********
<***********
***********>
********>
*****>
**>
```

Turtle Patterns

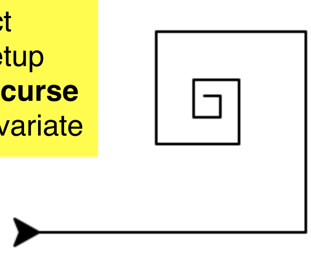
Invariate: a function, quantity, or property that remains unchanged when a specified transformation is applied.

Translation for Turtles:
The Turtle ends where it started

act
setup
recurse

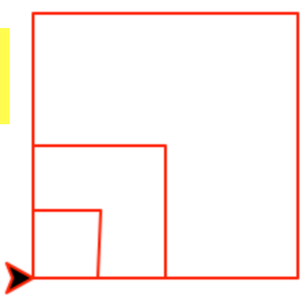


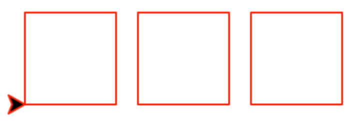
act
setup
recurse
invariate



The "setup"/"invariate" step is unique to Turtles; not needed in cs1graphics because there's no "traveling" of the graphics, everything is placed at a specific coordinate.

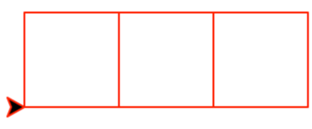
act
recurse

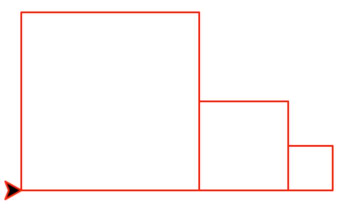


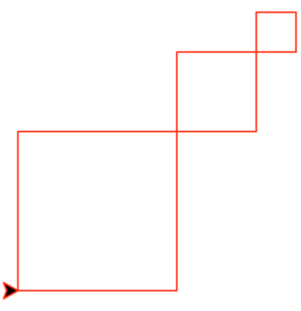


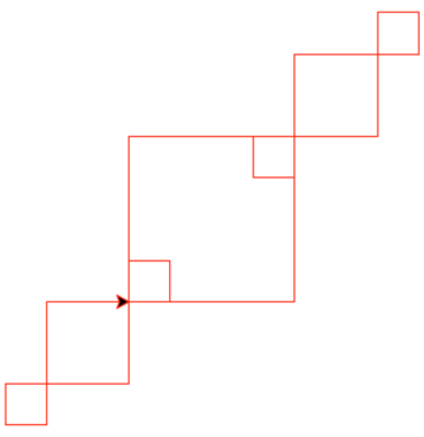
Design branches out in 1 direction.
As a result, there is 1 recursive call

act
setup
recurse
invariate









Design branches out in 2 directions
(North East, South West)

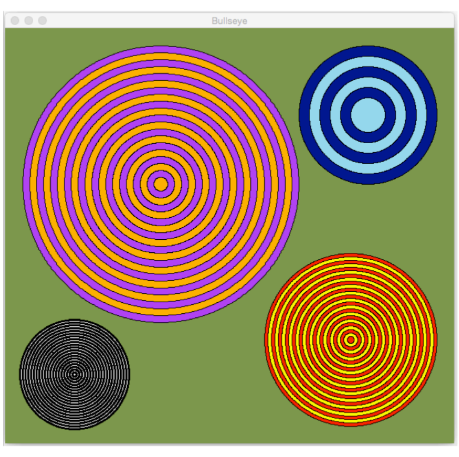
As a result, there are 2 recursive calls

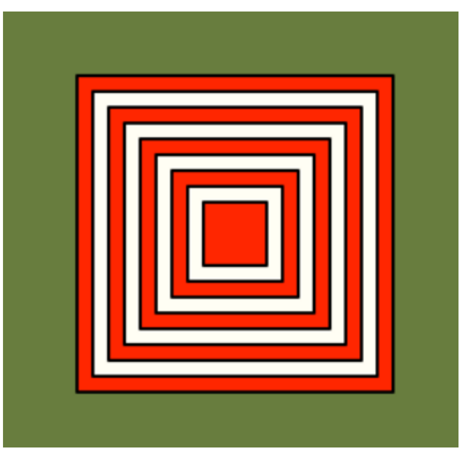
act
setup
recurse
invariate
recurse

Graphic Patterns

Design branches out in 1 direction (Inward)
As a result, there is 1 recursive call

act
recurse

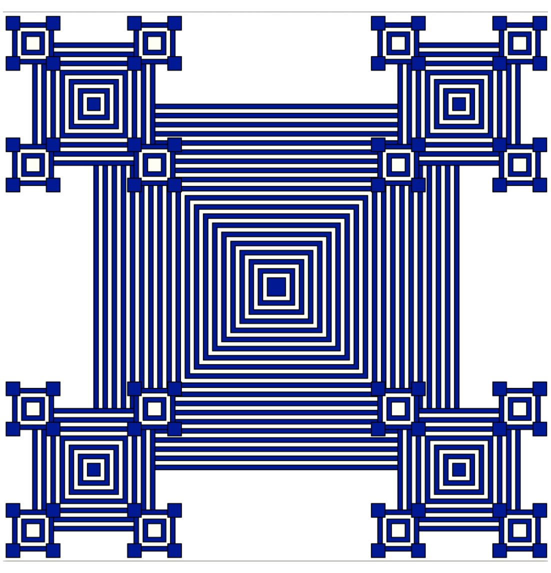




act
recurse
recurse
recurse
recurse

Design branches out in 4 directions
(North East, South East, South West, North West)

As a result, there are 4 recursive calls



Tip: Use a helper function to abstract the action and simplify the recursive function.
e.g. drawDisc, drawSquare