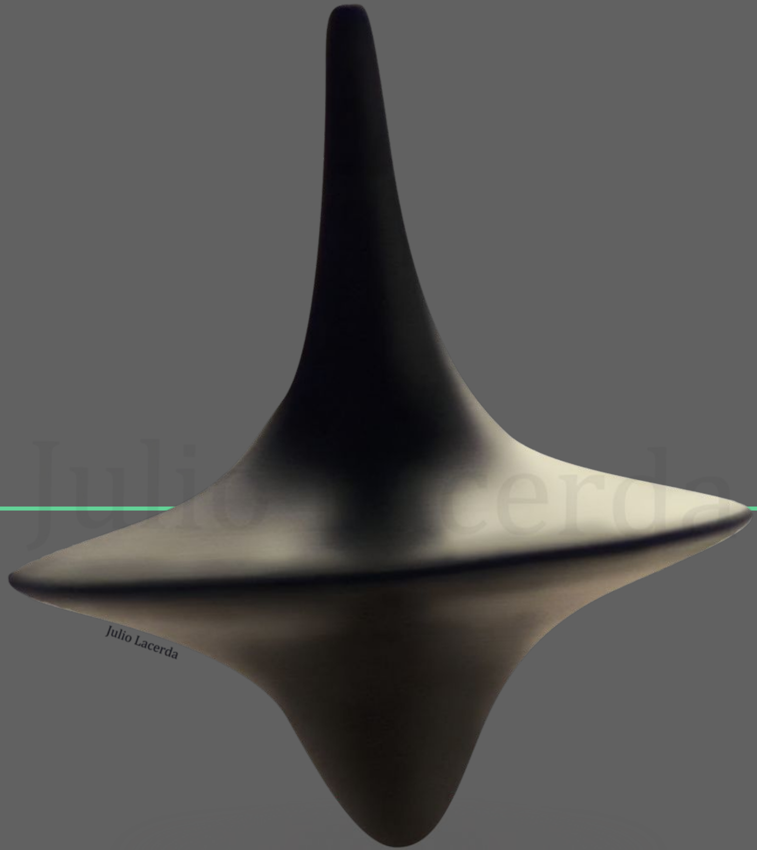
Recursion
Recursion
Recursion
Recursion
Recursion

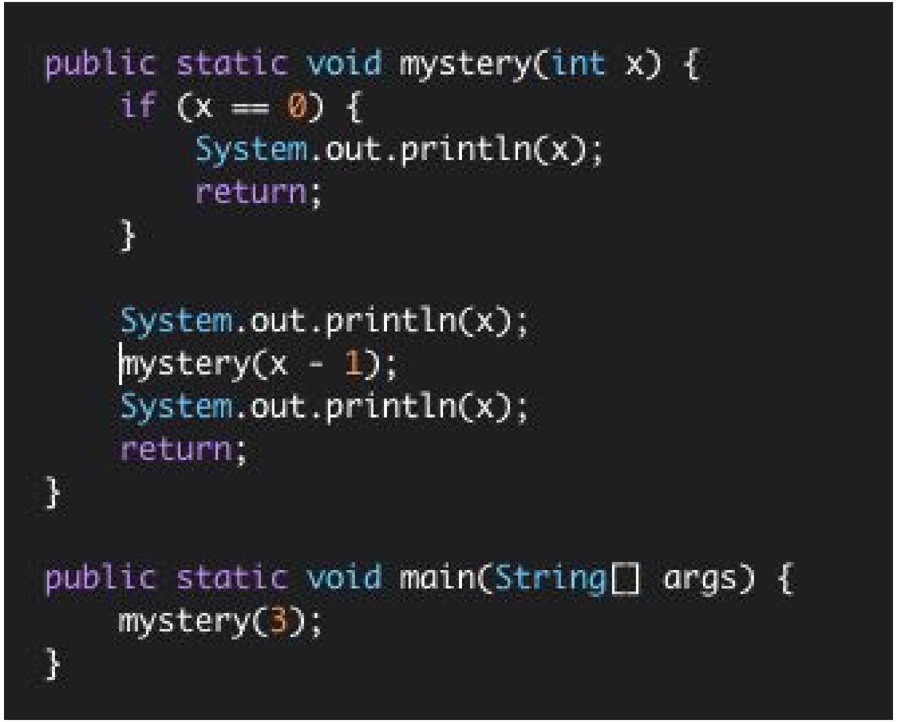


Cis 110 Recitation 7/19/17

Key concepts

- Recursive call(s): solve a smaller version of this problem and use that result
- Base case: Stop recursing when you get to a case that's simple enough to solve without recursion

Exercise: what does this print?

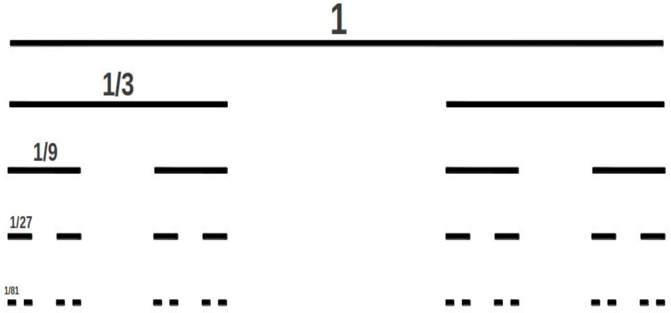


```java
public static void mystery(int x) {
    if (x == 0) {
        System.out.println(x);
        return;
    }

    System.out.println(x);
    mystery(x - 1);
    System.out.println(x);
    return;
}

public static void main(String[] args) {
    mystery(3);
}
```

Exercises:

For each of these, think about the base case and how you can break this problem into smaller parts.

- **public static int sumFirstN(int n)**
  - Return the sum 1 + 2 + ... + n
- **public static String reverseString(String input)**
  - Return the reversed version of `input`
  - Use `s.substring(i, j)`, which gives the characters in s starting with **`i`** and ending with **`j - 1`**
    - `“hello”.substring(0, “hello.length()) → “hello”`
    - `“hello”.substring(1, “hello.length() - 1) → “ell”`
- **public static void cantor(** /* whatever parameters you like */ **)**
  - Recursively draw this figure, called the Cantor set:


1
1/3
1/9
1/27
1/81