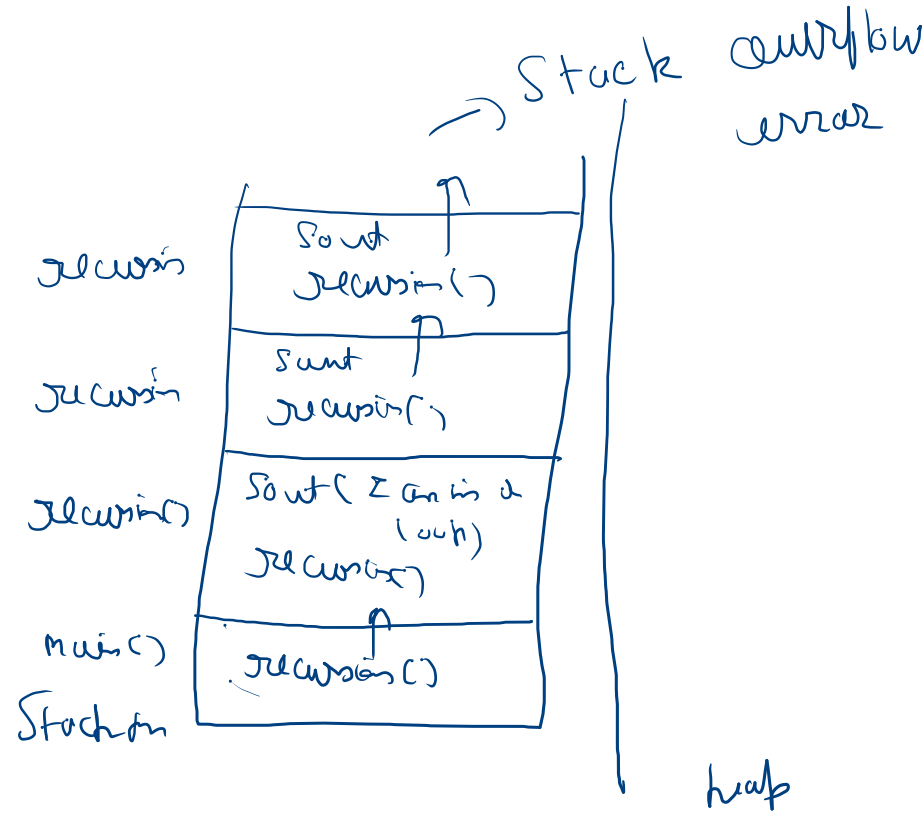


Recursion

→ A fn calls itself

2 usages

```
public static void recursion() {  
    System.out.println("I am in recursive loop");  
    recursion();  
}
```



1) Base condition

2 usages

```
public static void recursion(int n) {
```

Base condⁿ

```
if (n == 0) {
```

```
    return;
```

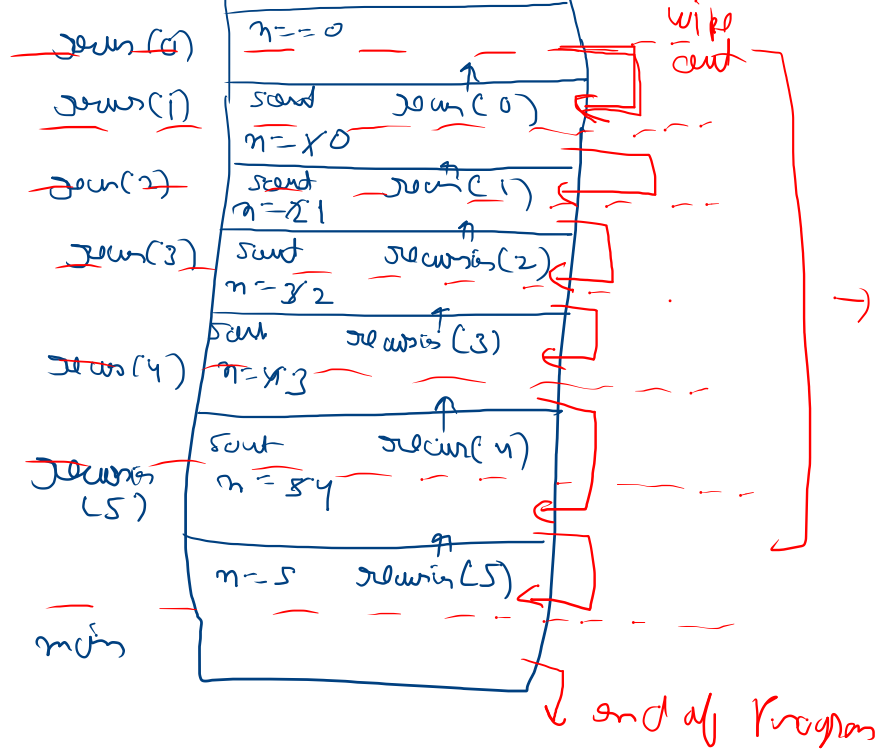
```
}
```

```
System.out.println(n);
```

```
n--;
```

```
recursion(n); → Lost  
end of code
```

→ Recursion and condition



5
4
3
2
1

Recursive
tree
==

$$\begin{array}{c} x(0) \\ \uparrow \\ x(1) \end{array} \left| \begin{array}{l} \text{ } \\ p \end{array} \right. \begin{array}{l} \text{ } \\ n = x_0 \end{array} \quad \left. \begin{array}{l} \text{ } \\ \text{ } \end{array} \right\}$$

$$\begin{array}{c} x(2) \\ \uparrow \\ p \end{array} \left| \begin{array}{l} \text{ } \\ \text{ } \end{array} \right. \begin{array}{l} \text{ } \\ n = x_1 \end{array}$$

$$\begin{array}{c} x(3) \\ \uparrow \\ p \end{array} \left| \begin{array}{l} \text{ } \\ \text{ } \end{array} \right. \begin{array}{l} \text{ } \\ n = x_2 \end{array}$$

$$\begin{array}{c} x(4) \\ \uparrow \\ p \end{array} \left| \begin{array}{l} \text{ } \\ \text{ } \end{array} \right. \begin{array}{l} \text{ } \\ n = x_3 \end{array}$$

$$\begin{array}{c} x(5) \\ \uparrow \\ p \end{array} \left| \begin{array}{l} \text{ } \\ \text{ } \end{array} \right. \begin{array}{l} \text{ } \\ n = x_4 \end{array}$$

~~$\text{rec}(0) \rightarrow 1, 2$~~
 \uparrow
 ~~$\text{rec}(1) \rightarrow 1, 2, 3, 4, 5, 6$~~ | $n=80$
 \uparrow
 $\text{rec}(2) \rightarrow 1, 2, 3, 4, 5, 6$ | $n=81$
 \uparrow
 $\rightarrow \text{rec}(3) \rightarrow 1, 2, 3, 4, 5, 6$ | $n=82$
 \uparrow
 $\text{rec}(4) \rightarrow 1, 2, 3, 4, 5, 6$ | $n=83$
 \uparrow
 $\text{rec}(5) \rightarrow 1, 2, 3, 4, 5, 6$ | $n=84$

```

1 System.out.println("Starting Recursive Stack " + n);
  // base condition
2 {
  if (n == 0) {
    return;
  }

3 System.out.println(n);
4 n--;     $n = n - 1 \rightarrow$  updating
5 recursion(n);
6 System.out.println("Ending Recursive Stack " + n);

```

\downarrow
 main

$x(1-1)$
↑

$n = 0$

$x(2-1)$
↑

$n = 1$

$x(3-1)$
↑

$n = 2$

$x(4-1)$
↑

$n = 3$

$x(5-1)$
↑

$n = 4$

$x(5) \rightarrow 1, 2, 3, 4$

$n = 5$

2 usages

```
public static void recursion(int n) {  
    1 System.out.println("Starting Recursive Stack " + n);  
    // base condition  
    2 { if (n == 0) {  
        return;  
    }  
    3 System.out.println(n);  
    n--;  
    4 recursion(n - 1);  
    5 System.out.println("Ending Recursive Stack " + n);  
}
```