

Recursion

↳ when a function calling itself

```
f() {  
    f();  
}
```

```
suman() {  
    suman();  
}
```

PMI

Sum of first n natural numbers = $\frac{n(n+1)}{2}$

① Prove for $n=1 \rightarrow \text{Sum} = 1$

② Assume for n

$$\sum_{i=1}^n i = \frac{n(n+1)}{2}$$

③ for $(n+1)$

$$\begin{aligned} & [1 + 2 + 3 + \dots + (n-1) + (n) + 1] \\ &= \frac{n(n+1)}{2} + (n+1) = \frac{(n+1)(n+2)}{2} \end{aligned}$$

Print Decreasing

Input $\rightarrow 5$

Output \rightarrow 5 4 3 2 1

```

    i = 0
    while (i < n) {
        printf("%d\n", i);
        i++;
    }
}

```

[]

Expectation

Faith

n	$n-1$
$n-1$	$n-2$
$n-2$	\vdots
\vdots	\vdots
3	3
2	2
1	1

- white (n) = L

$\text{void } \text{pd}(4) \{$
 $\quad \text{print}(\text{"pd } \dots");$
 $\quad \text{pd}(m+1);$
 $\}$

```
11  
12 void pd(int n){  
13     printf("%d\  
14     pd(n-1);  
15 }  
16
```

- 5
- 4
- 3
- 2
- 1

	$n = 1$	
pd	$n = 1$	① ②
pd	$n = 2$	① 2
bt	$n = 3$	① ②
pd	$n = 4$	① ②
pd	$n = 5$	① ②
main	$n = 5$	

```

11
12 void pd(int n){
13     if(n<1) return;
14     printf("%d\n",n);
15     pd(n-1);
16 }
17

```

	5	
	4	
	3	
	2	
	1	
pd	n = 0	(1)
pd	n = 1	(1) (2) (3)
pd	n = 2	(1) (2) (3)
pd	n = 3	(1) (2) (3)
pd	n = 4	(1) (2) (3)
pd	n = 5	(1) (2) (3)
main	n = 5	

Print Increasing

Input → 5
Output → 1 2 3 4 5

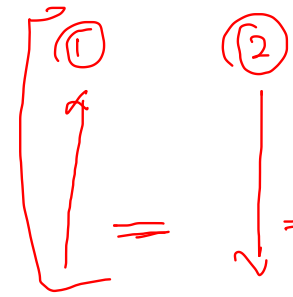
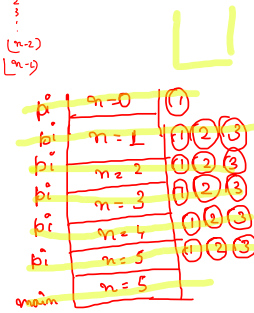
```
12 void pi(int n){
13     if(n<1) return; // ①
14     pi(n-1); // ②
15     printf("%d",n); // ③
16 }
17
18
19
```

pi(5) {

}

Emp
1
2
3
...
(n-2)
(n-1)
n

Faith
1
2
3
...
(n-2)
(n-1)
n



Print Decreasing Increasing

Input → 5

Output →
5
4
3
2
1
1
2
3
4
5

Factorial

Input → 5
Output → 120

Emp

Faith

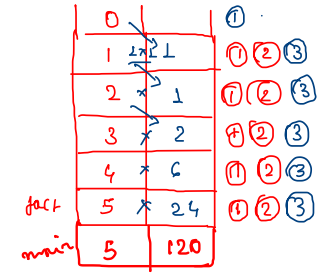
n x n → x ... L

[n-L x ... L]
k

Result → k * n

```
int fact(int n){
    if(n<1) return 1; // ①
    int k = fact(n-1); // ②
    return n*k; // ③
}
```

fact(5) {
 int k = fact(4);
 }



fact(5) → k * n = 120