

Recursion

① Find Factorial of Number N

⇒ [we are going to do enormous amount of questions in Recursion + DP]

⇒

•> Program to find factorial of number n

$$\Rightarrow \textcircled{n!} = \underline{n} \times \underline{(n-1)} \times \underline{(n-2)} \times \dots \times \underline{1}$$

$$\begin{aligned} \underline{\underline{\text{Ex}}} \quad \underline{\underline{5!}} &= \underline{5} \times \underline{4} \times \underline{3} \times \underline{2} \times \underline{1} \\ &= \underline{\underline{120}} \end{aligned}$$

$$\begin{aligned} \underline{\underline{\text{Ex}}} \quad \underline{\underline{4!}} &= \underline{4} \times \underline{3} \times \underline{2} \times \underline{1} \\ &= \underline{\underline{24}} \end{aligned}$$

Solve :

ex :

5!

↳ for (int i = n ; i >= 1 ; i--)

<

ans = ans x i

>

return

ans

↳ $1 \times 5 = \text{ans} = 5$
 $5 \times 4 = \text{ans} = 20$

int ans = 1

O(n)



Time

120

But
Now, onwards
we will think
of Recursive
solution

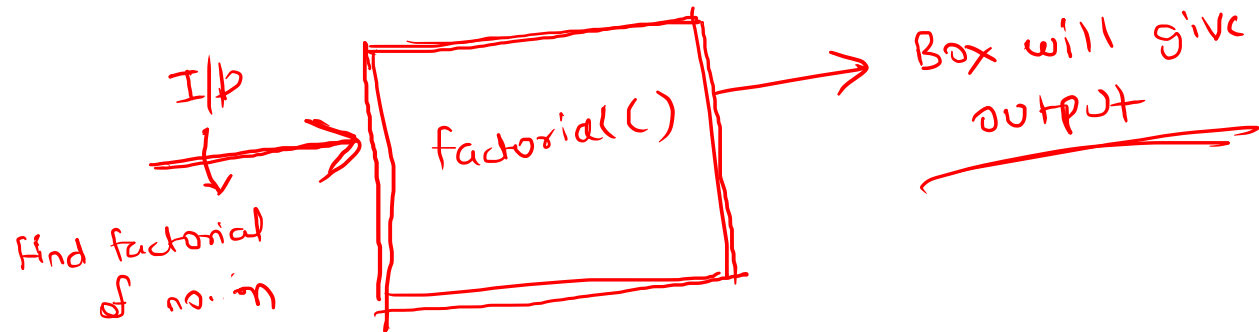
Ex : Find $n!$
 \Rightarrow Steps to create Recursive function

\bullet I/p \rightarrow a integer
(whose factorial
we have to find out)

o/p \rightarrow value of calculated
factorial

\downarrow
Let's suppose
function
name

factorial()





Now, try to use Magical Box

Now
Logic comes
in Picture

$\therefore \rightarrow$ this logic varies from problem to problem

$$\underline{\text{factorial}(5)} \rightarrow 5 \times \underline{(4 \times 3 \times 2 \times 1)} \quad \checkmark$$

$$\underline{\text{factorial}(4)} \rightarrow \underline{(4 \times 3 \times 2 \times 1)}$$

↳

Now we can do this

factorial(5) \rightarrow

5 \times

factorial(4)

this is given by
magical function

factorial

Format
of Recursive
function

void recursive (n)
{
 // _____ → Base condition
 recursive () → Recursive function call
 n-1
}

Recursive
sort to find
factorial

```
int factorial (5)  
{  
    // Base condn  
    return 5 * factorial (4)  
}
```

60%

```
int fact(n)  
{  
    // _____  
    return n * fact(n-1);  
}
```

✓


~~★~~ Sigma Rule

Base Condition

① Base condition also called as Termination condition ☹️

② To find the Base condition

↳ Try to find the output for the
possible input in the parameter.

Smallest] 

fact (int 2) →

0, 1, 2, 3 -
↑ ↑

Base
codⁿ #

```
int factorial(n)  
{
```

```
    return n * factorial(n-1);
```

```
}
```

→ Smallest Possible
Value of 'n'
↓ → 0, 1, 2, ...

if (n == 0) → o/p → 1

∴
if (n == 1) → o/p → 1



1 x 0 → 0
X

Base
Code #

```
int factorial(n)
{
    if (n == 0 || n == 1) return 1;
    return n * factorial(n-1);
}
```

→ Smallest Possible
Value of 'n'
↓ → 0, 1, 2, ...

if (n == 0) → o/p → 1
i.
if (n == 1) → o/p → 1

Recursive
function cell

```
⇒ int fact(n)  
< if(n==0 || n==1) return 1;  
> return n * fact(n-1);
```

if(n <= 1)
return 1

else #
return n * ~~fact(n-1)~~

fact(4) → n

4 × fact(3)

3 × fact(2)

2 × fact(1)

1
2

(24)

1 × fact(0)

0 × fact(-1)

-1000 X

4! = 24

```
fact(n)
{
  // Base → if (n <= 1) return 1
  return n × fact(n-1)
}
```