

🚩 Current Skill Study of an example (mathematic)

## Study of an Example:

Now, let's discuss a practical problem which can be solved by using recursion and understand its basic working and see how we apply our rules. An everyday example: factorial!

To calculate factorial of  $n$  we apply this formula :

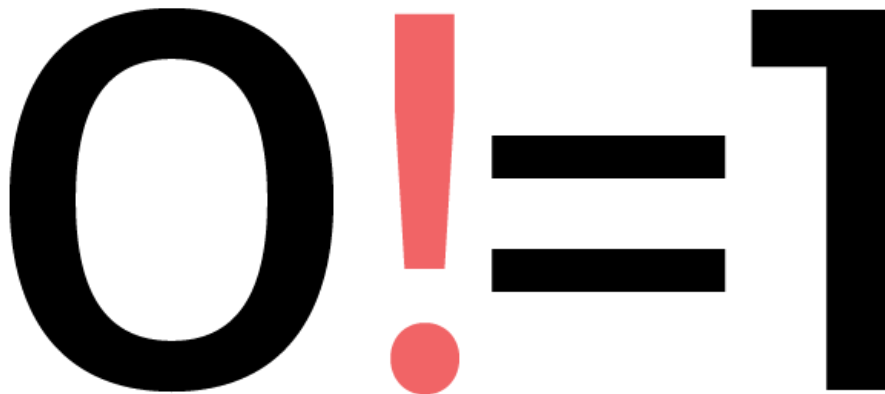
$$n! = n \times (n-1) \times (n-2) \times \dots \times 2 \times 1$$

So the first rule, what is the recursion condition here ? We can ask a question, "How we calculate factorial of  $n-1$  ?" that is right like this :  $n-1! = (n-1) \times (n-2) \times \dots \times 2 \times 1$

And that means that can we change the first formula now :  $n! = n \times (n-1)!$

We have here our recursion as we can do the same thing with  $n-1!$ ,  $n-2!$  ...

And remember


$$0! = 1$$

## Study of an Example:

The Second rule, what is the base case here ? We know that both factorial of 0 equals 1, this is our base case. Now, is this case can be enough as stop condition ?

Yes, because the recursion that we found is decreasing  $n$  in each call, and  $n$  must be null or

positive to calculate the factorials. This means that whatever  $n$  we choose to calculate factorials of will always converge to base case.

```
FUNCTION fact(n) : INTEGER
```

```
VAR
```

```
    results : INTEGER;
```

```
BEGIN
```

```
    IF (n = 0) THEN
```

```
        results := 1; // results of the factorial is 1
```

```
    ELSE
```

```
        results := n * fact(n-1); // n! = n * (n-1)!
```

```
    END_IF
```

```
    RETURN results ;
```

```
END
```

## Execution:

Calculation of 4! :

$\text{fact}(4) \Rightarrow 4 * \text{fact}(3) \Rightarrow 4 * 3 * \text{fact}(2) \Rightarrow 4 * 3 * 2 * \text{fact}(1) \Rightarrow 4 * 3 * 2 * 1 * \text{fact}(0) \Rightarrow 4 * 3 * 2 * 1 * 1$   
= 24

Examples of FALSE programs!

No stopping conditions  $\Rightarrow$  INFINITE CALCULATION



```
BEGIN
```

```
    results := n * fact(n-1);
```

```
    RETURN results ;
```

```
END
```

No simplification of the case to be treated

```
BEGIN
```

```
    IF (n = 0) THEN
```

```
        results := 1;
```

```
    ELSE
```

```
        results := n * fact(n+1);
```

```
    //the function is not directed to the stop condition.
```

```
    END_IF
```



RETURN results ;

END

< Previous

next >

