Recursión – Recursividad

Un subprograma puede llamar a cualquier otro subprograma y éste a otro, y así sucesivamente, es decir, los subprogramas se pueden anidar. Se puede tener el siguiente caso:

- A llamar_a B, B llamar_a C, C llamar_a D

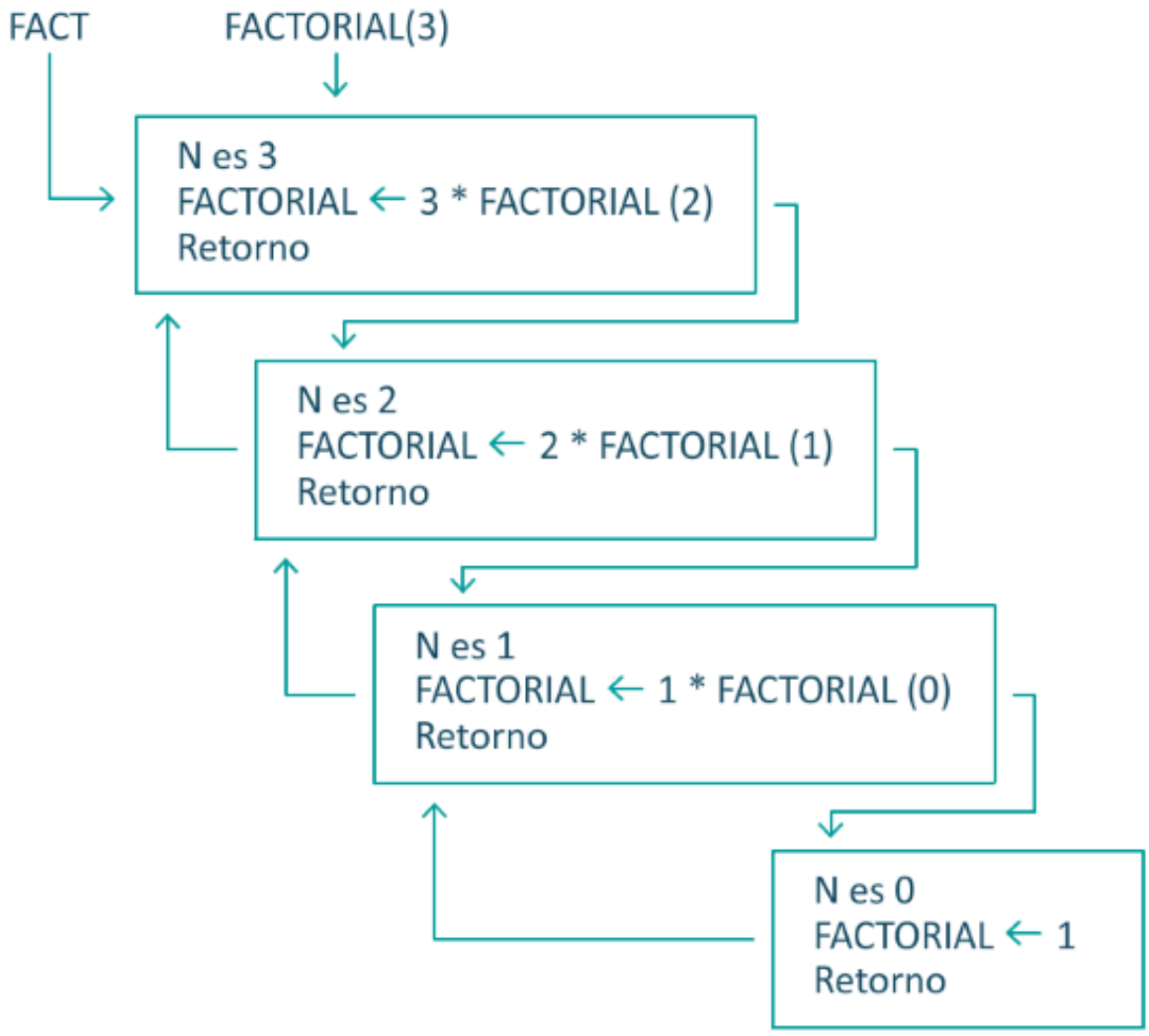
Cuando los subprogramas retornan al terminar cada uno de ellos el proceso resultante será:

- D retornar_a C, C retornar_a B, C retornar_a A

Sin embargo, en los lenguajes de programación un subprograma puede llamarse a sí mismo, en cuyo caso se dice que es recursivo. Esta recursividad es una herramienta muy potente en algunas aplicaciones, sobre todo de cálculo. La recursión puede ser utilizada como una alternativa a la iteración o repetición y su uso es particularmente idóneo para la solución de aquellos problemas que pueden definirse de modo natural en términos recursivos. Muchas funciones matemáticas se definen de forma recursiva. Un ejemplo es la función factorial.

$$n! = \begin{cases} 1 & \text{si} \quad n = 0 \\ n\,(n-1)! & \text{si} \quad n > 0 \end{cases}$$


```
def factorial (n)
```



Convenio de asociación no. 777 de 2021 suscrito entre el Fondo Único de Tecnologías de la Información y las Comunicaciones y la Universidad el Bosque.