

Problema 1.

Elaborar un función que calcule el factorial de un número entero positivo. Con el objeto de observar el proceso de ejecución recursivo, coloque una sentencia de escritura antes (“autollamada” i) y después de la llamada (“retorno de la llamada” i).

Función original:

func factorial(entero n): entero

inicio

si (n = 0) **entonces**

retornar(1)

sino

retornar(n * factorial(n-1))

fsi

ffunc // fin factorial

Función transformada:

func factorial(entero norig, entero n): entero

var

entero m

inicio

si (n = 0) **entonces**

retornar(1)

sino

para i ← 1 **hasta** norig – n **en** 1 **hacer**

escribir(" ") // dejar espacio en blanco

fpara

escribir("Autollamada con n = ", n)

m ← factorial(n-1)

para i ← 1 **hasta** norig – n **en** 1 **hacer**

escribir(" ") // dejar espacio en blanco

fpara

escribir("Retorno llamada n = ", n, " retornar(", n, "*", m, ") = retornar(", n*m, ")")

retornar(n * m) // originalmente **retornar**(n * factorial(n-1))

fsi

ffunc // fin factorial

Problema 2.

Elabore un función recursiva que dado un número real a y un entero no negativo b calcule a^b .

Problema 3.

Elabore una función recursiva que dado un entero no negativo n calcule su equivalente en binario.

Problema 4.

Haga otra versión de la función del problema 3 de tal manera que sea capaz de convertir el número entero no negativo n en su equivalente en cualquier otra base numérica.

Problema 5.

Dado un número entero positivo n , realice una función recursiva que permita calcular su cantidad de dígitos.