Laboratorio ~~# 3~~

Computabilidad y Complejidad de Algoritmo.

Fecha de Entrega: viernes 12 de noviembre 2019

1. Determina que calcula la siguiente función recursiva. Escribe una función interactiva que realice la misma tarea.

```
ALGORITMO N func(E N n)
VAR
     N sal
     INICIO
          SI n == 0 ENTONCES
               sal = 0
          SINO
               sal = n + func(n-1)
          FINSI
     DEVOLVER sal
FIN
```

2. Diseñar la versión iteractiva del siguiente algoritmo recursivo:

```
ALGORITMO Rec(E N n)
INICIO
     SI NO f(n) ENTONCES
          {cualquier grupo de sentencias que no modifiquen n}
           Rec(g(n))
      FINSI
FIN
```

Donde las cabeceras de `f` y `g` se definen como:

```
ALGORITMO B f(E N n)
ALGORITMO N g(E N n)
```

Nota:

Recordemos la composición de funciones,

Si $f(x)$ y $g(x)$ son funciones de una variable, podemos definir $h(x) = f \circ g(x)$ donde $f * g(x) = g(f(x))$

Ejemplo

- $f(x) = x - 5$ y $g(x) = x^2$, entonces
- $h(x) = f * g(x) = g(f(x)) = g(x - 5) = (x - 5)^2$

3. Considera la siguiente función recursiva:

```
ALGORITMO Z p(E Z x)
VAR Z result
INICIO
      SI x < 3 ENTONCES
            result = x
      SINO
            result = p(x-1) * p(x-3)
      FINSI
DEVOLVER result
FIN
```

Diseña una función recursiva que calcule el número de productos realizados al ejecutar la función `p` cuando se llama con el argumento `n`
.

4. Dada la función recursiva:

```
ALGORITMO Z f(E Z x)
VAR
Z result
INICIO
SI x>100 ENTONCES
result = x-10
SINO
result = f(f(x+11))
FINSI
DEVOLVER result
FIN
```

Estudia cuál es su comportamiento. ¿Podrías diseñar `f` de una manera más sencilla?

5. función de Ackermann

$$Ackerman(m,n) = \begin{cases} n+1 & si\ m=0 \\ Ackerman(m-1,1) & si\ m>0 \land n=0 \\ Ackerman(m-1, Ackerman(m,n-1)) & si\ m>0 \land n>0 \end{cases}$$

Esta función, llamada función de Ackermann, es interesante porque crece rápidamente con respecto a los valores `m` y `n`. Comprobar que `Ackermann(1, 2)` vale 4 y que `Ackermann(3, 2)` vale 29. ¿Cuantas llamadas recursivas se hacen cuando queremos evaluar `Ackermann(1,2)` ?