

Tema: Técnicas de Diseño de Algoritmos

1. Dadas las siguientes funciones misterio:

```
FUNCIÓN misterio1(n): entero $\geq$ 0  $\rightarrow$  entero $\geq$ 0
    SI (n < 10) ENTONCES
        Retorna n
    SINO
        Retorna resto(n,10) + misterio1(n/10)
```

```
FUNCIÓN misterio2(n): entero $\geq$ 0  $\rightarrow$  entero $>$ 0
    SI (n = 0) ENTONCES
        Retorna 1
    SINO
        Retorna 3*misterio2(n-1)
```

- Indique cuál es la tarea que lleva a cabo cada una de ellas.
 - Plantee la recurrencia, desarrolle y calcule el costo de las funciones en notación O grande.
- Escriba un procedimiento **recursivo** que, dado un número natural, escriba sus dígitos en orden inverso.
 - Diseñe una función que, dado un número entero positivo, devuelva el dígito de mayor valor en el número. Escriba una versión **iterativa** y otra **recursiva** de la misma.
Ej: N=1823 retorna 8
N=265 retorna 6
 - Diseñe una función **recursiva** que, dado un arreglo de n caracteres, devuelva verdadero si el arreglo contiene una palabra palíndrome.
 - Diseñe una función **recursiva** que reciba un arreglo de n números enteros y retorne verdadero si los elementos están ordenados en forma ascendente, caso contrario retorna falso.
 - Teniendo en cuenta la lista enlazada de números enteros implementada en el TP N° 1, diseñe una versión **recursiva** de la función **posicionK**.
 - Como usuario de la lista enlazada implemente una función **recursiva estaContenida** que, dadas dos listas enlazadas de números enteros, retorne verdadero si todos los elementos de la primera lista se encuentran presentes en la segunda, caso contrario retorna falso.
 - Dado un arreglo ordenado v de n números enteros, escriba un algoritmo D&C **buscaPar**, que compruebe si un par de enteros dados (x, y), distintos entre sí, ocupa posiciones consecutivas en el mismo. El método debe retornar verdadero si x e y son contiguos y falso en caso contrario.

En los ejercicios del 2 al 8:

- Calcule las complejidades de los algoritmos diseñados usando notación O grande.
- Implemente en código C++ y pruebe.