

## Guía Práctica 7: Funciones Recursivas

El proceso de la recursividad es el siguiente:

- Se llama a una función recursiva y se le envía un problema.
- La función analiza si puede resolver el problema (caso base), si es así lo resuelve y termina.
- Si no puede resolver el problema lo convierte en uno de menor complejidad y lo envía de nuevo a otra instancia de su función.
- El proceso se repite hasta que llega a un caso base que puede resolver, lo resuelve y termina.



**TIPS para desarrollar una función recursiva son:**

- Cada llamada recursiva (a la misma función) debe definir un problema de menor complejidad.
- Debe existir por lo menos un caso base (o será infinita!).

Para cada uno de los ejercicios propuestos a continuación, implementar una solución recursiva y dibujar la traza de ejecución para los ejemplos dados:

1. Escribir una función recursiva que calcule y retorne la *cantidad de dígitos* de un número entero.

Ejemplos de uso:

`cantDigitos(1234) → 4`

`cantDigitos(0) → 1`

Variantes:

- 1.1 Escribir una función recursiva que calcule la *cantidad de dígitos pares* de un número entero.

Ejemplos de uso:

`cantDigPares(1234) → 2`

`cantDigPares(44) → 2`

`cantDigPares(1) → 0`

`cantDigPares(4) → 1`

- 1.2 Escribir una función recursiva que indique si un número entero tiene *algún dígito par*.

Ejemplos de uso:

`tieneDigPares(1234) → true`

`tieneDigPares(44) → true`

`tieneDigPares(1) → false`

`tieneDigPares(4) → true`

- 1.3 Escribir una función recursiva que calcule la *suma de los dígitos* de un número entero que son *múltiplos de N*.

Ejemplos de uso:

`sumaDigMul(1234,2) → 6`

`sumaDigMul(44,4) → 8`

`sumaDigMul(7,3) → 0`

`sumaDigMul(4,4) → 4`

2. Escribir una función recursiva que devuelva la suma de los primeros N enteros.

Ejemplos de uso:

`suma(5) → 1+2+3+4+5 = 15`

`suma(1) → 1`

Variantes:

- 2.1 Escribir un programa que retorne la suma de los enteros positivos pares desde N hasta 2.

Ejemplos de uso:

`sumaPares(12) → 12+10+8+6+4+2= 42`

`sumaPares(11) → 10+8+6+4+2= 30`

`sumaPares(1) → 0`

`sumaPares(3) → 2`

3. Dado un número entero largo, imprimirlo en *orden inverso*. Resolverlo a través de una función recursiva

Ejemplos de uso:

`inverso(35891) → 1 9 8 5 3`

`inverso(7) → 7`

4. Escribir una función recursiva que nos diga *si un número es capicúa*.

Ejemplos de uso:

`capicua(35891) → false`

`capicua(35853) → true`

`capicua(7) → true`

5. Escribir una función que sea recursiva, con nombre *maxDigito(n)*, que reciba un número natural largo y devuelve el dígito más grande del número.

Ejemplos de uso:

`maxDigito(35897) → 9`

`maxDigito(7) → 7`

6. Determinar el *producto de dos números* a través de la suma y de manera recursiva.

7. Escribir una función recursiva que calcule la *potencia de un número*.

Ejemplos de uso:

`potencia(2,5) → 32`

`potencia(71,1) → 71`

`potencia(13,0) → 1`

8. Escribir una función recursiva que transforme un *número entero positivo a notación binaria*.  
(<https://es.convertbinary.com/decimal-a-binario/>)

Ejemplos de uso:

`decBinario(41) → 101001`

9. Escribir una función recursiva que transforme un número expresado *en notación binaria a un número entero*.

Ejemplos de uso:

`BinDec(101001) → 41`

10. Un granjero ha comprado una pareja de conejos para criarlos y luego venderlos. Si la pareja de conejos produce una nueva pareja cada mes y la nueva pareja tarda un mes más en ser también productiva, ¿cuántos pares de conejos podrá poner a la venta el granjero al cabo de un año?
11. Las *secuencias de ccanobif* son secuencias en las que cada término es siempre igual a la suma de los dos siguientes. Excepto por los dos últimos términos que siempre son iguales a 1. Dado un valor entero, imprimir la secuencia Iccanobif de tamaño correspondiente.

Ejemplos de uso:

`ccanobif(10) → 55 34 21 13 8 5 3 2 1 1 ccanobif(5) → 5 3 2 1 1`

12. Implementar una función recursiva *cuentaRegresiva(secuencia)* que informe una secuencia de despegue de una nave espacial, con el siguiente comportamiento:

- Si los segundos que faltan para el despegue (el único parámetro de la función) son menos que 10, se visualiza ese valor, se espera un segundo y se continúa. Si falta 0 segundos, se visualiza “DESPEGUE” y se termina.
- Si faltan más de 10 segundos, sólo hay que informar la cantidad de segundos que faltan para el despegue, cada 10 segundos, cuando el dígito unidad es 0 (y hay que esperar la cantidad de segundos que corresponda).

Para la espera utilizar la función Sleep(mseg) de la librería <windows.h>

Ejemplos de uso:

`cuentaRegresiva(35) --> 30 - 20 - 10 - 9 - 8 - 7 - 6 - 5 - 4 - 3 - 2 - 1 - DESPEGUE`

`cuentaRegresiva(6) --> 6 - 5 - 4 - 3 - 2 - 1 - DESPEGUE`

13. Dados dos números naturales  $n$  y  $b$  con  $b < n$ , podemos hacer que  $n$  explote usando a  $b$  como bomba. Cuando  $n$  explota se parte en dos números naturales  $n1 = (n \div b)$  y  $n2 = (n - n1)$  y se produce una reacción en cadena. Si  $n1$  (o  $n2$ ) es mayor que  $b$ , también explota y se parte nuevamente en dos pedazos, siguiendo el criterio anterior. Esto se repite hasta que todos los pedazos resultantes de la explosión de  $n$  sean menores o iguales que  $b$ . Diseña y escribe una función recursiva que, dados dos números  $n$  y  $b$ , muestre todos los pedazos de la explosión de  $n$  usando a  $b$  como bomba.

Ejemplos de uso:

Si  $n=23$  y  $b=3$ , los pedazos son 2 1 1 3 1 1 3 3 2 2 1 3 puesto que las explosiones generan: (7,16), luego cada uno ((2,5), (5,9)) y así sucesivamente.