



## Tema 4 - Recursión

### Hoja de Problemas

1. Implementa una función recursiva que reciba un número entero mayor o igual que cero y devuelva su factorial.
2. Implementa una función recursiva que devuelva la suma de los  $n$  primeros números naturales.
3. Implementa una función recursiva que reciba dos números enteros,  $a$  y  $n$ , siendo  $n$  mayor o igual que cero. La función devuelve la potencia de  $a$  elevado a  $n$ . No está permitido usar la función `pow`.
4. Implementa una función recursiva que reciba dos números enteros positivos,  $a$  y  $b$ , que devuelva el valor de la multiplicación de  $a$  por  $b$ . Pista:  $x \cdot y$  es igual que sumar el número  $x$ ,  $y$  veces.
5. Implementa una función recursiva que reciba dos números enteros positivos,  $a$  y  $b$ , y devuelva su máximo común divisor, aplicando el algoritmo de euclides.
6. Implementa una función que reciba una lista de Python, `data`, y devuelva la suma de sus elementos. Si la lista es nula (`None`), la función debe devolver `None`. Debes proponer tres versiones distintas de la función:
  - a. Permitido modificar el contenido de la lista.
  - b. No está permitido modificar el contenido de la lista pero es posible utilizar slicing.
  - c. No se permite modificar el contenido de `data` y debe optimizarse la complejidad espacial (es decir, no está permitido utilizar técnicas de slicing para considerar sublistas de la lista de entrada).
7. Implementa una función que reciba una lista de Python, `data`, y devuelva el mayor de sus elementos. Si la lista es nula (`None`) o está vacía, la función debe devolver `None`. Debes proponer tres versiones distintas de la función:
  - a. Permitido modificar el contenido de la lista.
  - b. No está permitido modificar el contenido de la lista pero es posible utilizar slicing.
  - c. No se permite modificar el contenido de `data` y debe optimizarse la complejidad espacial (es decir, no está permitido utilizar técnicas de slicing para considerar sublistas de la lista de entrada).

8. Implementa una función recursiva que reciba un string y compruebe si la palabra es palíndroma o no. Debes proponer dos versiones distintas (primera usando slicing y la segunda optimizando la complejidad espacial).
9. Implementa una función recursiva que reciba una array (lista de python) de enteros y compruebe si está ordenado. Debes proponer dos versiones distintas (la primera usando slicing y la segunda optimizando la complejidad espacial).
10. Implementa una función recursiva que reciba un array (lista de python) ordenado de enteros y un número entero, y compruebe si el número está en el array. En ese caso, la función devuelve True y en otro caso False. La solución debería ser lo más eficiente posible (en terminos de complejidad espacial).
11. Implementa una función recursiva que reciba un array (lista de python) ordenado de enteros sin repeticiones y un número entero, y devuelva la posición del número en el array. Si el número no existe, la función debe devolver -1. La solución debería ser lo más eficiente posible (en terminos de complejidad espacial).
12. Implementa una función recursiva que reciba un array (lista de python) de enteros y lo invierta. La función no devuelve nada, pero sí modifica el array. Implementa dos versiones distintas:
  - a. Una donde se permita añadir o eliminar elementos de la lista
  - b. Una donde no se permita añadir o eliminar elementos de la lista.
13. Implementa una función recursiva que reciba un número entero positivo y devuelve el número de dígitos que forman dicho número. No puedes tratar el número como un string o una lista de Python.
14. Implementa una función recursiva que reciba un número entero positivo y devuelve la suma de sus dígitos. No puedes tratar el número como un string o una lista de Python.
15. Implementa una función que dado un número natural devuelva su suma de fibonacci.
16. Implementa una función recursiva que reciba dos números enteros positivos, a y b, y devuelva su multiplicación aplicando el método de la multiplicación rusa.

37	6	37	6
18	12	<del>18</del>	<del>12</del>
9	24	9	24
4	48	<del>4</del>	<del>48</del>
2	96	<del>2</del>	<del>96</del>
1	192	1	192
			222

17. Implementa una función que reciba la ruta de un directorio y devuelva el tamaño que ocupa en disco.
18. Implementa una función recursiva que reciba un string y devuelve el número de vocales que tiene.