

# RELACIÓN DE PROBLEMAS EN JAVA

## (Recursión)

1. Potencia de un Número: Diseñar un método recursivo que calcule la potencia de un número real  $x$  elevado a un exponente entero  $n$ .
2. Suma de los  $N$  Primeros Números Naturales: Crear un método recursivo que calcule la suma de los primeros  $N$  números naturales.
3. 9. División por Restas Sucesivas: Implementar un algoritmo recursivo que permita realizar la división entre dos números usando restas sucesivas.
4. Longitud de una Cadena: Implementa una función recursiva que calcule la longitud de una cadena, sin utilizar métodos predefinidos de longitud. Esta función debe dividir la cadena en partes más pequeñas hasta llegar a la base del problema.
5. Eliminar Vocales de una Cadena: Diseña una función recursiva que elimine todas las vocales de una cadena. Por ejemplo, dado el string "recursión", la salida debería ser "rcrsón".
6. Suma de los Dígitos de un Número: Escribir un algoritmo recursivo que sume los dígitos de un número. Ejemplo: Entrada 123, Resultado 6.
7. Conteo de Vocales en una Cadena: Diseña un método recursivo que cuente el número de vocales en una cadena dada. Esto te ayudará a comprender cómo recorrer una cadena de forma recursiva y aplicar condiciones específicas.
8. Máximo Común Divisor (MCD): Programar un método recursivo que calcule el máximo común divisor de dos números usando el algoritmo de Euclides.
9. Número Combinatorio: Diseñar un método recursivo para calcular el número combinatorio de  $m$  elementos tomados de  $n$  en  $n$ , usando la ley de recurrencia. Este problema se resuelve utilizando la fórmula del coeficiente binomial.
10. Buscar en un Array Ordenado: Crear un método recursivo que busque un número en un array y devuelva su posición o -1 si no está presente.
11. Permutaciones de una Cadena: Crea una función recursiva que genere todas las permutaciones posibles de una cadena dada. Este problema es útil para aprender a manejar problemas donde hay que generar múltiples combinaciones de elementos.
12. Sumar Dígitos Pares: Escribe una función recursiva que sume todos los dígitos pares de un número. Esto requiere descomponer el número y comprobar cada dígito.
13. Generación de Combinaciones Binarias: Escribe un método recursivo para generar todas las combinaciones binarias posibles de una longitud  $n$ . Esto es útil para problemas relacionados con la teoría de la información y la generación de todos los posibles estados binarios.
14. Inversión de una Cadena: Crea una función recursiva que invierta una cadena. Por ejemplo, para la entrada "hola", la salida debería ser "aloh".