

Introducción a Java. Boletín II

1. Realizar un método llamado `numeroSolucionesEcuacionSegundoGrado` que reciba los coeficientes de una ecuación de segundo grado y devuelva el número de soluciones que tiene. Si los argumentos no son válidos (el primer coeficiente tiene que ser distinto de cero) debe devolver un -1.
2. Realiza un programa que pida un número por teclado y que después muestre ese número al revés.
3. Diseña una función que, dada una cadena de entrada, comprueba si es una contraseña **fuerte** o no. Se considera que una contraseña es fuerte si contiene 8 o más caracteres, y entre ellos al menos hay una mayúscula, una minúscula, un signo de puntuación y al menos un dígito.
4. Elabora un programa que codifique una cadena, de tal modo que en el resultado se inviertan cada 2 caracteres. Los caracteres intercambiados no pueden volver a intercambiarse. Ejemplo:

Entrada -> Hola mundo
Salida -> oHalm nuod
5. Realizar un método llamado **esMultiplo** que recibirá dos números y devuelva True si el primer número es múltiplo del segundo.
6. Realizar un método llamado `horaMayor` que recibirá seis valores enteros, los tres primeros representarán la hora, minuto y segundos de la primera hora y los otros tres de la segunda hora. Se deberá devolver un 1 si la primera hora es mayor que la segunda, un 2 si la segunda hora es mayor que la primera, un 0 si son iguales y un -1000 si los datos no son correctos.
7. Realizar un método llamado `segundosEntre` que recibirá seis valores enteros, los tres primeros representarán la hora, minuto y segundos de la primera hora y los otros tres de la segunda hora. Se deberá devolver el número de segundos que hay entre la primera hora y la segunda (el valor debe ser siempre en positivo). Si los datos no son correctos se deberá devolver -1000.

8. Realiza un método llamado **toBinary** que reciba un número decimal como argumento y devuelva un String con el número binario correspondiente.
9. Realiza un método llamado **toDecimal** que reciba un String con un valor numérico en binario como argumento y devuelva un número con el número decimal correspondiente.
10. Realiza un método llamado **gcd** (greaterCommonDivisor) que recibirá dos números y devuelva el máximo común divisor según el algoritmo de Euclides.

Algorithm Greatest Common Divisor (I).

Input. Two positive integers a and b greater than zero.

Output. The greatest common divisor (a, b) .

1. While $(b > 0)$ do
 - 1.1 $r \leftarrow a \pmod{b}$
 - 1.2 $a \leftarrow b$
 - 1.3 $b \leftarrow r$
 2. Return(a).
-

$$\begin{aligned} 1220 \bmod 516 &= 188 \\ 516 \bmod 188 &= 140 \\ 188 \bmod 140 &= 48 \\ 140 \bmod 48 &= 44 \\ 48 \bmod 44 &= 4 \\ 44 \bmod 4 &= 0 \\ 4 &= \text{GCD} \end{aligned}$$

11. Realizar un método llamado **minimoComunMultiplo** que reciba dos números y calcule el mínimo común múltiplo de dos números. Con el máximo común divisor de una pareja de números podemos obtener fácilmente el mínimo común múltiplo de dicha pareja. El mínimo común múltiplo de dos números es igual al producto de los números dividido entre su máximo común divisor. Por ejemplo, el máximo común divisor de 24 y 36 es 12, por tanto el mínimo común múltiplo de 24 y 36 es $(24 \times 36) / 12 = 72$.