Ejercicios 2.3 Funciones

Ejercicio 1

Realiza una función que reciba la base y el exponente y devuelva la potencia $base^{exponente}$ sin utilizar Math.pow.

Ejercicio 2

Realiza una función que reciba 3 parámetros: dos de tipo entero y uno de tipo ENUM. La función deberá sumar, restar, multiplicar o dividir los valores de los dos primeros parámetros dependiendo de la operación indicada en el tercer parámetro, y devolver el resultado.

Ejercicio 3

Sobrecarga la función del ejercicio anterior para que se pueda operar con enteros y con decimales. Haz un programa que utilice las dos funciones, con enteros y con decimales.

Ejercicio 4

Realiza una función que encuentre el primer valor n para el que la suma `1 + 2 + 3 + ... + n` exceda a un valor m que se introduce por parámetro. Es decir, si m vale:

- 1: devuelve 2
- 3: devuelve 3
- 7: devuelve 4
- 10: devuelve 5
- 15: devuelve 6

Ejercicio 5

El máximo común divisor (mcd) de dos enteros es el entero más grande que es divisor exacto de los dos números. Realiza una función que devuelva el máximo común divisor de dos enteros. Por ejemplo, 12 es el mcd de 36 y 60.

Ejercicio 6

Se dice que un número entero es primo si solo es divisible entre 1 y entre sí mismo. Por ejemplo, 2, 3, 5 y 7 son primos, pero 4, 6, 8 y 9 no lo son.

1. Realiza una función que devuelva si un número es primo o no.
2. Realiza una función que muestre todos los números primos entre 1 y 10.000.
3. Realiza una función que descomponga un número en factores primos. Ejemplos:
    - 18 = 2 * 3 * 3
    - 11 = 11
    - 35 = 5 * 7
    - 40 = 2 * 2 * 2 * 5

Ejercicio 7

Se dice que un número entero es un número perfecto si la suma de sus divisores propios (incluyendo el 1 y sin incluirse él mismo) da como resultado el mismo número. Por ejemplo, 6 es un número perfecto, porque sus divisores propios son 1, 2 y 3; y 6 = 1 + 2 + 3. Los siguientes números perfectos son 28, 496 y 8128.

1. Realiza una función que devuelva si el parámetro es perfecto o no.
2. Realiza una función que dado un número perfecto, imprima los factores para confirmar que el número es perfecto. Si no lo es, que no haga nada. Utiliza la función creada en el apartado 1.
3. Realiza una función que muestre todos los números perfectos entre 1 y 10.000 con sus correspondientes factores. Utiliza la función creada en el apartado 1.

Ejercicio 8

En matemáticas, la sucesión o serie de Fibonacci es la siguiente sucesión infinita de números naturales:

0, 1, 1, 2, 3, 5, 8, 13, 21, 34, ...

La serie comienza con los números 0 y 1 y a partir de éstos, cada término es la suma de los dos anteriores.

1. Realiza una función que devuelva el elemento enésimo de la serie de Fibonacci. Es decir, si recibe:
    - 0: devuelve 0
    - 1: devuelve 1
    - 4: devuelve 3
    - 7: devuelve 13
2. Realiza una función que muestre los 30 primeros números de la serie de Fibonacci. Utiliza la función creada en el apartado 1.
3. Realiza una función que calcule el primer elemento de la serie de Fibonacci que sea mayor o igual que un valor introducido por parámetro. Por ejemplo, si recibe 20, devolverá 21 ya que es el primer elemento de la serie mayor o igual que 20. Utiliza la función creada en el apartado 1.

Ejercicio 9

Realiza una función que reciba un número entero positivo de n cifras y devuelva el número con sus cifras en orden inverso. No utilizar String ni calcular previamente el número de cifras. Ej: 54.321 debe devolver 12.345.