



Full Stack

Fundación Telefónica Movistar
CURSO DE INTRODUCCIÓN A FULL STACK.

Guía de ejercicios.

Ejercicio 1

Escribir un programa que, a partir de dos números enteros, muestre por pantalla todos los números pares que hay entre ambos números, incluyendo, en caso de ser pares, los números originales.

Ejercicio 2

Asignar un número cualquiera, entero mayor que cero a una variable.

Verificar si el número ingresado es múltiplo de 2, 3, 4, 5, 6, 7, 8 o 9.

Armar un arreglo con los números encontrados (por ejemplo, si el número ingresado es múltiplo de 3, 6 y 7, armamos un arreglo que contenga estos tres números).

Mostrar el arreglo por pantalla, ordenado de mayor a menor.

En caso de que el número ingresado no sea múltiplo de estos números, mostrar por pantalla el mensaje “No se encontraron divisores exactos”.

Ejercicio 3

Crear un arreglo con 10 números enteros y mostrarlos por pantalla.

Luego reemplazar todos los números pares por la palabra PAR usando una estructura FOR y volver a mostrar el arreglo por pantalla.

Ejercicio 4

Dada una medida de tiempo expresada en horas, minutos y segundos con valores arbitrarios, elabore un programa que transforme dicha medida en una expresión correcta.

Por ejemplo, dada la medida 3h 118m 195s, el programa deberá obtener como resultado 5h 1m 15s.

Ejercicio 5

Escriba un programa que muestre en pantalla todos los números primos entre 1 y n, donde n es un número positivo que cargamos en una variable al inicio.



Ejercicio 6

Crea una función llamada **calculadora** que tome tres argumentos: dos números y una cadena que represente una operación (por ejemplo, "+", "-", "*", "/"). La función debe realizar la operación y devolver el resultado.

Ejercicio 7

Crea una función llamada **contarPalabras** que tome una cadena de texto como argumento y devuelva el número de palabras en esa cadena. Las palabras están separadas por espacios.

Ejercicio 8

Crea una función llamada **generarContraseña** que genere una contraseña aleatoria de una longitud especificada. La contraseña debe contener letras mayúsculas, letras minúsculas y números.

Ejercicio 9

Crea una función llamada **validarCorreo** que tome una dirección de correo electrónico como argumento y determine si es una dirección de correo electrónico válida.

Ejercicio 10

Crea una función llamada **calcularPromedio** que tome un array de números como argumento y devuelva el promedio de esos números.

