



Github



Linkedin



Youtube



Rust

Tema

Funciones

Capítulo 4.

Ejercicios Tema

1. Diseñar la función (fun) `eco (num: i32)` a la que se le pasa como parámetros un número `n`, y muestra por pantalla `n` veces el mensaje:

Eco ...

2. Escribir una función a la que se le pasen dos enteros y muestre todos los números comprendidos entre ellos.
3. Realizar una función (fun) que calcule y muestre el área o el volumen de un cilindro, según se especifique. Para distinguir un caso de otro se le pasará un número 1 (para área) o 2 (para el volumen).

Además, hemos de pasarle a la función el radio de la base y la altura.

4. Diseñar una función que reciba como parámetros dos números enteros y que devuelva el máximo (el mayor de los dos).
5. Repetir el ejercicio anterior con una versión que calcule el máximo de 3 números.
6. Crear una función que, mediante un booleano, indique si el carácter que se pasa como parámetro de entrada corresponde con una vocal.

7. Diseñar una función que nos diga si un número es primo.
8. Escribir una función (fun) a la que se le pase un número entero y devuelva el número de divisores primos que tiene.
9. Implementar la función (fun) divisores_primos() que muestra, por consola, todos los divisores primos del número que se le pasa como parámetro.

10. Escribir una función (fun) que decida si dos números enteros positivos son amigos.

Dos números son amigos si la suma de sus divisores propios (distintos de ellos mismos) son iguales entres si.

Es decir, $\text{sum_divisores_propios de } a = b$ y viceversa, $\text{sum_divisores_propios de } b = a$

11. Diseñar una función (fun) que calcule a^n , donde a es real y n entero no es negativo.

Realizar una versión iterativa y otra recursiva.

12. Diseñar la función (fun) calculadora (num1: i32, num2: i32), a la que se pasan dos números enteros (operandos) y qué operación se desea realizar con ellos. Las operaciones disponibles son: sumar, restar, multiplicar o dividir.

Estas se especifican mediante un número: 1 para la suma, 2 para la resta, 3 para la multiplicación y 4 para la división.

13. Calcular el factorial de n recursivamente. Recordar que por definición el factorial de 0 es (0!) es 1.

14. Diseñar una función que calcule el nésimo término de la serie de Fibonacci. En esta serie el nésimo valor se calcula sumando los dos valores anteriores. Es decir:

$\text{fibonacci} (n) = \text{fibonacci} (n - 1) + \text{fibonacci} (n - 2)$

$\text{fibonacci} (0)$

$\text{fibonacci} (1)$



Github



Linkedin



Youtube

