

## **GUÍA #3 - EJERCICIOS DE RECURSIÓN**

### **Ejercicio 1:**

Escriba una definición recursiva de una función que tiene un parámetro  $n$  de tipo entero y que devuelve el  $n$ -ésimo número de Fibonacci. Los números de Fibonacci se definen de la siguiente manera:

$$F_0 = 1$$

$$F_1 = 1$$

$$F_{i+2} = F_i + F_{i+1}$$

### **Ejercicio 2**

La forma para calcular cuantas maneras diferentes tengo para elegir  $r$  cosas distintas de un conjunto de  $n$  cosas es:

$$C(n,r) = n! / (r! * (n-r)!)$$

Donde la función factorial se define como

$$n! = n * (n-1) * (n-2) * \dots * 2 * 1$$

Descubra una versión recursiva de la fórmula anterior y escriba una función recursiva que calcule el valor de dicha fórmula.

### **Ejercicio 3**

Escriba una función recursiva que ordene de menor a mayor un arreglo de enteros basándose en la siguiente idea: coloque el elemento más pequeño en la primera ubicación, y luego ordene el resto del arreglo con una llamada recursiva.

### **Ejercicio 4**

Escribir una función recursiva que devuelva la suma de los primeros  $N$  enteros

### **Ejercicio 5**

Escribir un programa que encuentre la suma de los enteros positivos pares desde  $N$  hasta 2. Chequear que si  $N$  es impar se imprima un mensaje de error.

### **Ejercicio 6**

Escribir un programa que calcule el máximo común divisor (MCD) de dos enteros positivos. Si  $M \geq N$  una función recursiva para MCD es

$$\text{MCD} = M \text{ si } N = 0$$

$$\text{MCD} = \text{MCD}(N, M \bmod N) \text{ si } N > 0$$

El programa le debe permitir al usuario ingresar los valores para  $M$  y  $N$  desde la consola. Una función recursiva es entonces llamada para calcular el MCD. El programa entonces imprime el valor para el MCD. Si el usuario ingresa un valor para  $M$  que es  $<$  que  $N$  el programa es responsable de switchear los valores.

### **Ejercicio 7**

Programa un método recursivo que transforme un número entero positivo a notación binaria.

#### **Ejercicio 8**

Programa un método recursivo que transforme un número expresado en notación binaria a un número entero.

#### **Ejercicio 9**

Programa un método recursivo que calcule la suma de un arreglo de números enteros.

#### **Ejercicio 10**

Programa un método recursivo que invierta los números de un arreglo de enteros.

#### **Ejercicio 12**

Implemente una función recursiva que nos diga si una cadena es palíndromo.

#### **Ejercicio 13**

Implementa una función recursiva que calcule la potencia de 'x' elevada al valor 'y'.

#### **Ejercicio 14**

Escribe una función recursiva que realice un conteo regresivo desde un número dado hasta 0.

#### **Ejercicio 15**

Implementa una función recursiva que invierta una cadena dada.

#### **Ejercicio 16**

Crea una función recursiva que cuente la cantidad de dígitos en un número entero positivo.

#### **Ejercicio 17**

Diseña una función recursiva que calcule la suma de los dígitos en un número entero.

#### **Ejercicio 18**

Crea una función recursiva que realice una búsqueda binaria en un arreglo ordenado.

#### **Ejercicio 19**

Escribe una función recursiva que genere todas las permutaciones posibles de un conjunto de elementos.

#### **Ejercicio 20**

Implementa una función recursiva que determine si es posible obtener una suma específica utilizando elementos de un arreglo.

#### **Ejercicio 21**

Diseña una función recursiva para verificar si un número dado es primo.

#### **Ejercicio 22**

Diseña una función recursiva que calcule el número de formas diferentes en que se pueden dar cambios usando un conjunto de monedas.

**Ejercicio 23**

Escribe una función recursiva que encuentre todos los factores primos de un número entero.

**Ejercicio 24**

Implementa una función recursiva que recorra una matriz (arreglo bidimensional) e imprima sus elementos.

**Ejercicio 25**

Implementa una función recursiva que convierta un número en base 10 a otra base dada.

**Ejercicio 26**

Crea una función recursiva que calcule el número de formas posibles de subir una escalera con 'n' escalones, tomando uno o dos pasos a la vez.

**Ejercicio 27**

Implementa una función recursiva que genere todos los subconjuntos de un conjunto dado.

**Ejercicio 28**

Diseña una función recursiva que calcule la suma de todos los factores de un número entero positivo.

**Ejercicio 29**

Diseña una función recursiva que encuentre todos los números cuyas cifras son todas diferentes en un rango dado.

**Ejercicio 30**

Calcula el valor de 'e' (número de Euler) usando una serie infinita mediante una función recursiva.