

PREUNIVERSITARIO

Práctico 3: Estructuras repetitivas

Objetivo:

Implementar ciclos para resolver problemas que requieran repetición de acciones.

Competencias a desarrollar:

- ✓ Diseño y Desarrollo de Algoritmos Eficientes: El estudiante será capaz de diseñar y desarrollar algoritmos utilizando estructuras de control repetitivas (PARA, MIENTRAS, REPETIR-MIENTRAS) para resolver problemas matemáticos y de lógica, aplicando correctamente operaciones matemáticas y cálculos.
- ✓ Escritura Eficaz de Pseudocódigo y Documentación: El estudiante podrá escribir pseudocódigo de manera estructurada y clara, utilizando comentarios para explicar el funcionamiento de cada parte del algoritmo.
- ✓ Interacción con el Usuario y Validación de Datos: El estudiante será capaz de diseñar programas que interactúen con el usuario, solicitando datos de entrada y proporcionando resultados claros y concisos.

Te sugerimos primero resolver los ejercicios por tu cuenta y luego comparar tus respuestas con las soluciones propuestas. Recuerda que no existe una única forma de resolver un problema.

Ejercicios

- 1) Crea un programa que imprima en pantalla todos los números enteros desde 0 hasta 100 (incluyendo ambos extremos), en orden creciente, mostrando un número por línea.
- 2) Desarrolla un programa que solicite al usuario un número entero y determine la cantidad de dígitos que contiene.
- 3) Escribe un programa que sume todos los números enteros comprendidos entre dos valores dados por el usuario, excluyendo esos dos valores.
- 4) Elabora un programa que permita al usuario ingresar números enteros y los sume en secuencia. El programa debe detenerse y mostrar el total acumulado cuando el usuario ingrese un 0.
- 5) Crea un juego en el que el usuario deba adivinar un número aleatorio entre 0 y 9. Al final, el programa debe mostrar cuántos intentos fueron necesarios para acertar el número.
- 6) Desarrolla un programa que imprima en pantalla todos los números pares comprendidos entre 0 y 100, en orden decreciente.

- 7) Crea un programa que calcule la suma de todos los números comprendidos entre 0 y un número entero positivo indicado por el usuario.
- 8) Escribe un programa que permita al usuario ingresar 100 números enteros. Luego, el programa debe indicar cuántos de estos números son pares, cuántos son impares, cuántos son negativos y cuántos son positivos. (Nota: para probar el programa puedes usar una cantidad menor, pero debe estar preparado para procesar 100 números con un solo cambio).
- 9) Elabora un programa que permita al usuario ingresar 100 números enteros y luego calcule la media de esos valores. (Nota: puedes probar el programa con una cantidad menor, pero debe poder procesar 100 números cambiando solo un valor).
- 10) Desarrolla un programa que calcule el factorial de un número entero dado por el usuario. Ejemplo: para el número 5, el resultado debe ser $5! = 5 \times 4 \times 3 \times 2 \times 1$.
- 11) Crea un programa que permita al usuario ingresar 100 números y determine cuál de esos valores es el mayor y cuál es el menor. (Nota: para probar el programa puedes usar una cantidad menor, pero debe estar listo para procesar 100 números con un solo cambio).
- 12) Desarrolla un programa que determine si un número ingresado por el usuario es primo (es decir, que solo tenga dos divisores: 1 y el propio número). El número 1 debe considerarse como no primo.
- 13) Escribe un programa que encuentre el dígito más pequeño de un número entero ingresado por el usuario.
- 14) Crea un programa que imprima en pantalla todos los números pares entre 0 y 100, en orden inverso.
- 15) Desarrolla un programa que calcule la suma de todos los números enteros comprendidos entre 0 y un número natural ingresado por el usuario.
- 16) Escribe un programa que invierta el orden de los dígitos de un número ingresado por el usuario. Ejemplo: si el usuario ingresa 547, el programa debe mostrar 745.
- 17) Crea un programa que verifique si un número ingresado por el usuario es un número perfecto. Un número perfecto es un número positivo que es igual a la suma de sus divisores positivos, excluyendo el mismo número.
- 18) Desarrolla un programa para calcular el Máximo Común Divisor (MCD) entre dos números ingresados por el usuario. Usa cinco variables para este cálculo: 'num1' y 'num2' para almacenar los números ingresados, 'menor' para el menor de los dos números, 'mcd' para almacenar el resultado y 'i' como variable de iteración en el bucle.