

Guía de ejercicios

Ejercicio 1

Enunciado:

Diseña un programa que reciba una cantidad de dinero e indique cuántos billetes de 50 000, 20 000, 10 000, 5 000, 2 000 y 1 000 se necesitan para completar esa suma, utilizando la menor cantidad posible de billetes.

Competencia que evalúa:

Uso de operadores aritméticos (división y módulo) y aplicación de condicionales para resolver problemas de descomposición numérica.

Ejercicio 2

Enunciado:

Crea un programa que solicite el peso (kg) y la altura (m) de una persona, calcule el **Índice de Masa Corporal (IMC)** y muestre un mensaje según el resultado:

- Menor a 18.5: Bajo peso
- Entre 18.5 y 24.9: Normal
- Entre 25 y 29.9: Sobrepeso
- 30 o más: Obesidad

Competencia que evalúa:

Aplicación de condicionales anidados y operadores relacionales para clasificar datos numéricos.

Ejercicio 3

Enunciado:

Realiza un programa que solicite tres números al usuario y determine cuál es el **mayor** y cuál es el **menor**.

Competencia que evalúa:

Uso de condicionales compuestos y operadores lógicos para comparar múltiples valores.

Ejercicio 4

Enunciado:

Desarrolla un programa que solicite al usuario el valor de una compra y calcule el total a pagar aplicando un **descuento del 10%** si el valor supera los \$100 000.

Competencia que evalúa:

Comprensión de operadores aritméticos y condicionales simples para aplicar reglas de negocio.

Ejercicio 5

Enunciado:

Crea un programa que lea la edad de una persona y determine si puede votar (mayor o igual a 18 años), si debe presentar tarjeta de identidad (entre 7 y 17 años), o si es menor para tener documento.

Competencia que evalúa:

Uso de estructuras `if`, `elif`, `else` para manejar condiciones múltiples con intervalos numéricos.

Ejercicio 6

Enunciado:

Escribe un programa que simule un intento de acceso con contraseña. El usuario tendrá **tres intentos** para escribir la clave correcta. Si la acierta antes de agotar los intentos, el programa muestra un mensaje de éxito; si no, muestra "Acceso bloqueado".

Competencia que evalúa:

Control de ciclos con contador y validación de condiciones combinadas.

Ejercicio 7

Enunciado:

Crea un programa que pida al usuario un número y muestre si es **positivo**, **negativo** o **igual a cero**.

Competencia que evalúa:

Aplicación de operadores relacionales y condicionales para clasificación numérica.

Ejercicio 8

Enunciado:

Diseña un programa que solicite al usuario un número entero y muestre su **tabla de multiplicar** del 1 al 10.

Competencia que evalúa:

Uso de ciclos **for** o **while** para generar secuencias controladas.

Ejercicio 9**Enunciado:**

Elabora un programa que pida al usuario un número y determine si es **par** o **impar**.

Competencia que evalúa:

Uso del operador módulo (%) y condicionales simples para validación aritmética.

Ejercicio 10**Enunciado:**

Realiza un programa que muestre todos los números del 1 al 100 y al final indique cuántos de ellos son múltiplos de 3.

Competencia que evalúa:

Uso de ciclos con contador, operaciones aritméticas y condicionales para acumulación de resultados.

Ejercicio 11**Enunciado:**

Crea un programa que lea tres números diferentes y determine si pueden formar los lados de un triángulo. Si es posible, indica qué tipo de triángulo es:

- Equilátero (tres lados iguales)
- Isósceles (dos lados iguales)
- Escaleno (todos diferentes)

Competencia que evalúa:

Uso de condicionales anidados, operadores lógicos y validación de relaciones numéricas para clasificar casos múltiples.

Ejercicio 12**Enunciado:**

Desarrolla un programa que calcule el salario final de un trabajador.

El usuario ingresa:

- Horas trabajadas

- Valor por hora
Si trabaja más de 40 horas, las horas extras se pagan al **150%** del valor normal.

Competencia que evalúa:

Aplicación de estructuras condicionales y operadores aritméticos para resolver un problema real con condiciones de cálculo variables.

Ejercicio 13

Enunciado:

Elabora un programa que solicite un número entero positivo y determine si es **número primo** (solo divisible por 1 y por sí mismo).

Competencia que evalúa:

Uso de ciclos con control lógico y operadores módulo para validar divisibilidad y optimizar procesos de decisión.

Ejercicio 14

Enunciado:

Diseña un programa que muestre los primeros **n términos de la serie de Fibonacci**, donde **n** es un número entero ingresado por el usuario.

Competencia que evalúa:

Aplicación de ciclos con variables acumuladoras y comprensión de relaciones recursivas sin uso de funciones.

Ejercicio 15

Enunciado:

Crea un programa que simule una calculadora con un **menú interactivo** que permita realizar suma, resta, multiplicación y división. El programa debe repetirse hasta que el usuario elija la opción “Salir”.

Competencia que evalúa:

Uso de bucles controlados por condición, estructuras de selección múltiple y manejo de menús interactivos básicos.

Ejercicio 16

Enunciado:

Escribe un programa que pida un número entero y calcule la **suma de todos los números pares** desde 1 hasta ese número.

Competencia que evalúa:

Uso de ciclos y operadores lógicos para acumulación condicional dentro de un rango numérico.

Ejercicio 17**Enunciado:**

Crea un programa que calcule la **potencia** de un número **base** elevado a un exponente **n** sin usar el operador ******. Debes usar un ciclo para realizar el cálculo multiplicativo.

Competencia que evalúa:

Construcción de algoritmos iterativos con acumuladores y comprensión de multiplicación sucesiva controlada.

Ejercicio 18**Enunciado:**

Desarrolla un programa que lea un número entero y determine la **cantidad de dígitos** que tiene.

Competencia que evalúa:

Uso de divisiones enteras y ciclos **while** para análisis numérico y control de iteraciones según valor residual.

Ejercicio 19**Enunciado:**

Crea un programa que simule un contador regresivo desde un número **n** hasta 0, mostrando un mensaje al final que diga “¡Despegue!”.

Competencia que evalúa:

Aplicación de ciclos **while** con decremento controlado y manejo de condiciones de salida.

Ejercicio 20**Enunciado:**

Escribe un programa que solicite una cantidad **n** y calcule el **factorial** de ese número (**n!**), es decir, el producto de todos los números enteros desde 1 hasta **n**.

Competencia que evalúa:

Implementación de ciclos con acumuladores y comprensión de patrones matemáticos iterativos.

Ejercicio 21

Enunciado:

Crea un programa que solicite dos números enteros y determine si el segundo es **múltiplo exacto** del primero. Si no lo es, el programa debe indicar cuántas unidades faltan para que lo sea.

Competencia que evalúa:

Uso combinado de operadores módulo, condicionales y aritmética para verificar y calcular relaciones numéricas.

Ejercicio 22

Enunciado:

Desarrolla un programa que pida al usuario un número entero positivo **n** y muestre todos los números del 1 al **n**, indicando junto a cada uno si es par o impar.

Competencia que evalúa:

Uso de ciclos con validación condicional y salida controlada para análisis secuencial de valores numéricos.

Ejercicio 23

Enunciado:

Crea un programa que calcule la **suma de los dígitos** de un número entero positivo (por ejemplo, $1234 \rightarrow 10$). No se permite convertir el número a texto.

Competencia que evalúa:

Manipulación numérica con divisiones enteras y módulo, y uso de bucles controlados por condición para descomposición aritmética.

Ejercicio 24

Enunciado:

Elabora un programa que solicite un número entero positivo y determine si es un **número perfecto**, es decir, si la suma de sus divisores propios (excluyendo el número mismo) es igual al número.

Competencia que evalúa:

Implementación de ciclos anidados, divisibilidad, acumuladores y validación de condiciones lógicas complejas.

Ejercicio 25

Enunciado:

Desarrolla un programa que calcule la **media aritmética** de una cantidad

desconocida de números ingresados por el usuario. El proceso termina cuando el usuario ingrese el número 0, el cual no debe incluirse en el cálculo.

Competencia que evalúa:

Control de flujo con condición de salida, manejo de acumuladores y contadores, y validación de entrada dinámica.

Ejercicio 26

Enunciado:

Diseña un programa que permita calcular el **mínimo común múltiplo (MCM)** de dos números enteros positivos mediante sumas sucesivas.

Competencia que evalúa:

Comprensión de operaciones matemáticas iterativas, control de bucles y razonamiento lógico para alcanzar una condición de coincidencia.

Ejercicio 27

Enunciado:

Crea un programa que lea dos números enteros positivos **a** y **b** y calcule su **máximo común divisor (MCD)** utilizando el algoritmo de Euclides (mediante restas sucesivas).

Competencia que evalúa:

Aplicación de algoritmos clásicos, ciclos con condiciones dependientes y optimización de iteraciones lógicas.

Ejercicio 28

Enunciado:

Desarrolla un programa que genere un patrón numérico en forma de triángulo rectángulo. El usuario ingresa un número **n** y el programa imprime líneas del 1 hasta **n**, repitiendo el número según la línea.

Ejemplo para **n = 4**:

```
1
22
333
4444
```

Competencia que evalúa:

Uso de ciclos anidados (**for** o **while**) y control de variables para construcción de patrones lógicos numéricos.

Ejercicio 29

Enunciado:

Escribe un programa que pida un número entero y determine si es un **número Armstrong**. Un número es Armstrong si la suma de sus dígitos elevados al número de dígitos es igual al mismo número (por ejemplo, $153 \rightarrow 1^3 + 5^3 + 3^3 = 153$).

Competencia que evalúa:

Razonamiento aritmético avanzado, control de bucles, uso del operador potencia y validación de igualdad compleja.

Ejercicio 30

Enunciado:

Crea un programa que permita calcular el total a pagar en un parqueadero. El usuario ingresa las horas de permanencia y el tipo de vehículo:

- Carro: \$5 000 por hora
- Moto: \$2 000 por hora
Si la permanencia supera las 10 horas, se aplica un **descuento del 10%** sobre el total.

Competencia que evalúa:

Uso combinado de condicionales anidados, operadores aritméticos y estructuras de decisión para resolver un problema de negocio con múltiples condiciones.