[

{ "descripcion": "Escribe un programa que imprima 'Hola Mundo'.", "salida\_esperada": "Hola Mundo" },

{ "descripcion": "Escribe un programa que solicite al usuario su nombre y lo salude.", "salida\_esperada": "Hola, [nombre]!" },

{ "descripcion": "Escribe un programa que calcule la suma de dos números ingresados por el usuario.", "salida\_esperada": "La suma es [resultado]" },

{ "descripcion": "Escribe un programa que determine si un número ingresado es par o impar.", "salida\_esperada": "[número] es par." o "[número] es impar." },

{ "descripcion": "Escribe un programa que calcule el factorial de un número entero positivo.", "salida\_esperada": "El factorial de [número] es [resultado]" },

{ "descripcion": "Escribe un programa que imprima los números del 1 al 100 usando un bucle.", "salida\_esperada": "1 2 3 4 ... 100" },

{ "descripcion": "Escribe un programa que cuente cuántas vocales tiene una palabra ingresada.", "salida\_esperada": "La palabra [palabra] tiene [número] vocales." },

{ "descripcion": "Escribe un programa que determine si una palabra ingresada es un palíndromo.", "salida\_esperada": "[palabra] es un palíndromo." o "[palabra] no es un palíndromo." },

{ "descripcion": "Escribe un programa que ordene tres números ingresados por el usuario de mayor a menor.", "salida\_esperada": "[número mayor] [número intermedio] [número menor]" },

{ "descripcion": "Escribe un programa que muestre la tabla de multiplicar de un número ingresado.", "salida\_esperada": "La tabla de multiplicar de [número] es: 1x[número], 2x[número], ..., 10x[número]" }

]

[

{ "descripcion": "Escribe un programa que calcule el promedio de una lista de números ingresados por el usuario.", "salida\_esperada": "El promedio es [promedio]" },

{ "descripcion": "Escribe un programa que genere los primeros N números de la serie de Fibonacci.", "salida\_esperada": "0, 1, 1, 2, 3, ..., N números" },

{ "descripcion": "Escribe un programa que convierta grados Celsius a Fahrenheit.", "salida\_esperada": "[temperatura en Fahrenheit]" },

{ "descripcion": "Escribe un programa que determine si un número es primo.", "salida\_esperada": "[número] es primo." o "[número] no es primo." },

{ "descripcion": "Escribe un programa que invierta una cadena de texto ingresada por el usuario.", "salida\_esperada": "[cadena invertida]" },

{ "descripcion": "Escribe un programa que simule una calculadora con las operaciones básicas (suma, resta, multiplicación, división).", "salida\_esperada": "El resultado de la operación es [resultado]" },

{ "descripcion": "Escribe un programa que cuente cuántas veces aparece una letra específica en una frase.", "salida\_esperada": "La letra [letra] aparece [número] veces." },

{ "descripcion": "Escribe un programa que valide si una contraseña cumple con ciertos requisitos (mínimo 8 caracteres, una mayúscula, un número).", "salida\_esperada": "La contraseña es válida." o "La contraseña no es válida." },

{ "descripcion": "Escribe un programa que determine si un año ingresado es bisiesto.", "salida\_esperada": "[año] es bisiesto." o "[año] no es bisiesto." },

{ "descripcion": "Escribe un programa que simule un cajero automático con opciones de consulta, depósito y retiro.", "salida\_esperada": "Seleccione una opción: 1. Consulta 2. Depósito 3. Retiro" }

]

[

{ "descripcion": "Escribe un programa que encuentre el número mayor en un arreglo de enteros.", "salida\_esperada": "El número mayor es [número]" },

{ "descripcion": "Escribe un programa que elimine los elementos duplicados de un arreglo.", "salida\_esperada": "[arreglo sin duplicados]" },

{ "descripcion": "Escribe un programa que ordene un arreglo de números usando el algoritmo de burbuja.", "salida\_esperada": "[arreglo ordenado]" },

{ "descripcion": "Escribe un programa que implemente una búsqueda binaria en un arreglo ordenado.", "salida\_esperada": "El número [número] se encuentra en el índice [índice]" o "El número [número] no está en el arreglo." },

{ "descripcion": "Escribe un programa que cuente cuántas palabras hay en una oración ingresada por el usuario.", "salida\_esperada": "La oración tiene [número] palabras." },

{ "descripcion": "Escribe un programa que simule el lanzamiento de un dado y muestre el resultado.", "salida\_esperada": "El número obtenido es [número]" },

{ "descripcion": "Escribe un programa que determine si dos cadenas son anagramas.", "salida\_esperada": "[cadena1] y [cadena2] son anagramas." o "[cadena1] y [cadena2] no son anagramas." },

{ "descripcion": "Escribe un programa que convierta un número decimal a binario.", "salida\_esperada": "El número binario es [binario]" },

{ "descripcion": "Escribe un programa que lea 10 números y calcule cuántos son positivos, negativos o cero.", "salida\_esperada": "[número positivos] positivos, [número negativos] negativos, [número ceros] ceros" },

{ "descripcion": "Escribe un programa que implemente un menú interactivo usando `switch` para realizar diferentes operaciones matemáticas.", "salida\_esperada": "Elija una opción: 1. Suma 2. Resta 3. Multiplicación 4. División" }

]

[

{ "descripcion": "Escribe un programa que implemente una función recursiva para calcular la potencia de un número.", "salida\_esperada": "La potencia de [número] elevado a [exponente] es [resultado]" },

{ "descripcion": "Escribe un programa que genere una matriz 3x3 con números aleatorios y calcule su suma total.", "salida\_esperada": "La suma total de la matriz es [suma]" },

{ "descripcion": "Escribe un programa que calcule la media, mediana y moda de una lista de números.", "salida\_esperada": "Media: [media], Mediana: [mediana], Moda: [moda]" },

{ "descripcion": "Escribe un programa que implemente un sistema básico de autenticación con nombre de usuario y contraseña.", "salida\_esperada": "Bienvenido [usuario]!" o "Credenciales incorrectas." },

{ "descripcion": "Escribe un programa que simule una agenda de contactos con opción de agregar, buscar y eliminar.", "salida\_esperada": "[contacto agregado/buscado/eliminado]" },

{ "descripcion": "Escribe un programa que determine si una matriz cuadrada es simétrica.", "salida\_esperada": "La matriz es simétrica." o "La matriz no es simétrica." },

{ "descripcion": "Escribe un programa que calcule el determinante de una matriz 2x2 ingresada por el usuario.", "salida\_esperada": "El determinante de la matriz es [determinante]" },

{ "descripcion": "Escribe un programa que simule el juego de adivinar un número aleatorio entre 1 y 100.", "salida\_esperada": "Adivinaste el número en [intentos] intentos!" },

{ "descripcion": "Escribe un programa que convierta un número binario a decimal sin usar métodos integrados.", "salida\_esperada": "El número decimal es [decimal]" },

{ "descripcion": "Escribe un programa que implemente el algoritmo de ordenamiento por selección.", "salida\_esperada": "[arreglo ordenado]" }

]

[

{ "descripcion": "Escribe un programa que lea un archivo de texto y muestre su contenido línea por línea.", "salida\_esperada": "[contenido del archivo línea por línea]" },

{ "descripcion": "Escribe un programa que cree un archivo de texto y escriba una lista de nombres en él.", "salida\_esperada": "Archivo creado exitosamente." },

{ "descripcion": "Escribe un programa que implemente una clase `Persona` con atributos nombre, edad y métodos para mostrar su información.", "salida\_esperada": "Nombre: [nombre], Edad: [edad]" },

{ "descripcion": "Escribe un programa que implemente una clase `CuentaBancaria` con métodos para depositar, retirar y mostrar saldo.", "salida\_esperada": "Saldo actual: [saldo]" },

{ "descripcion": "Escribe un programa que use herencia para definir una clase `Estudiante` que extienda a `Persona` con un atributo adicional `nota`.", "salida\_esperada": "Nombre: [nombre], Edad: [edad], Nota: [nota]" },

{ "descripcion": "Escribe un programa que use una interfaz `Operaciones` con métodos sumar, restar, multiplicar y dividir, e implemente una clase que los defina.", "salida\_esperada": "Resultado: [resultado]" },

{ "descripcion": "Escribe un programa que maneje una lista de objetos `Producto` con atributos nombre y precio, y muestre los productos ordenados por precio.", "salida\_esperada": "[lista de productos ordenados]" },

{ "descripcion": "Escribe un programa que implemente una pila (stack) utilizando un arreglo.", "salida\_esperada": "[elemento] apilado/desapilado." },

{ "descripcion": "Escribe un programa que implemente una cola (queue) utilizando una lista enlazada.", "salida\_esperada": "[elemento] encolado/desencolado." },

{ "descripcion": "Escribe un programa que use excepciones para manejar errores al dividir entre cero.", "salida\_esperada": "Error: división entre cero." }

]

[

{ "descripcion": "Escribe un programa que implemente una búsqueda lineal en un arreglo de enteros.", "salida\_esperada": "El elemento [elemento] se encuentra en el índice [índice]." o "El elemento no está en el arreglo." },

{ "descripcion": "Escribe un programa que implemente una clase `Rectángulo` con métodos para calcular área y perímetro.", "salida\_esperada": "Área: [área], Perímetro: [perímetro]" },

{ "descripcion": "Escribe un programa que use polimorfismo para calcular el área de figuras: círculo, rectángulo y triángulo.", "salida\_esperada": "Área: [área]" },

{ "descripcion": "Escribe un programa que simule un sistema de inventario con opción de agregar, eliminar y buscar productos.", "salida\_esperada": "[producto agregado/eliminado/encontrado]" },

{ "descripcion": "Escribe un programa que implemente el algoritmo de ordenamiento quicksort.", "salida\_esperada": "[arreglo ordenado]" },

{ "descripcion": "Escribe un programa que use un `HashMap` para contar la frecuencia de palabras en una cadena de texto.", "salida\_esperada": "{ palabra1: frecuencia, palabra2: frecuencia }" },

{ "descripcion": "Escribe un programa que simule una agenda telefónica usando un `HashMap` para almacenar nombres y teléfonos.", "salida\_esperada": "[nombre]: [teléfono]" },

{ "descripcion": "Escribe un programa que cree una interfaz `Figura` con un método `calcularArea`, e implemente clases concretas para varias figuras.", "salida\_esperada": "Área: [área]" },

{ "descripcion": "Escribe un programa que serialice un objeto `Empleado` en un archivo y luego lo deserialice.", "salida\_esperada": "Objeto serializado/deserializado correctamente." },

{ "descripcion": "Escribe un programa que implemente una lista enlazada simple y permita insertar y eliminar nodos.", "salida\_esperada": "[lista enlazada resultante]" }

]

[

{ "descripcion": "Escribe un programa que implemente una lista doblemente enlazada con inserción y eliminación en ambos extremos.", "salida\_esperada": "[lista doblemente enlazada resultante]" },

{ "descripcion": "Escribe un programa que implemente una estructura de árbol binario de búsqueda con inserción y recorrido inorden.", "salida\_esperada": "[recorrido inorden del árbol]" },

{ "descripcion": "Escribe un programa que implemente una cola prioritaria utilizando una estructura de montículo (heap).", "salida\_esperada": "[elemento] insertado/eliminado." },

{ "descripcion": "Escribe un programa que implemente el algoritmo de búsqueda en profundidad (DFS) en un grafo representado como lista de adyacencia.", "salida\_esperada": "[orden de visita de nodos]" },

{ "descripcion": "Escribe un programa que implemente el algoritmo de búsqueda en anchura (BFS) en un grafo representado como matriz de adyacencia.", "salida\_esperada": "[orden de visita de nodos]" },

{ "descripcion": "Escribe un programa que lea datos desde un archivo CSV y los almacene en una lista de objetos.", "salida\_esperada": "[lista de objetos leídos]" },

{ "descripcion": "Escribe un programa que conecte a una base de datos SQLite y permita insertar y consultar datos desde una tabla `Usuarios`.", "salida\_esperada": "[datos insertados/consultados]" },

{ "descripcion": "Escribe un programa que implemente hilos (`Threads`) para imprimir números pares e impares de forma concurrente.", "salida\_esperada": "Pares: [pares], Impares: [impares]" },

{ "descripcion": "Escribe un programa que implemente el patrón de diseño Singleton.", "salida\_esperada": "Instancia única creada." },

{ "descripcion": "Escribe un programa que calcule la expresión matemática de una cadena (como '3 + 5 \* 2') respetando la jerarquía de operaciones.", "salida\_esperada": "Resultado: [resultado]" }

]

[

{ "descripcion": "Escribe un programa que implemente una interfaz gráfica simple usando Swing para sumar dos números ingresados por el usuario.", "salida\_esperada": "Resultado: [resultado]" },

{ "descripcion": "Escribe un programa que use sockets para establecer una comunicación simple entre un cliente y un servidor.", "salida\_esperada": "Mensaje recibido: [mensaje]" },

{ "descripcion": "Escribe un programa que implemente una calculadora científica con interfaz gráfica usando JavaFX.", "salida\_esperada": "Resultado: [resultado]" },

{ "descripcion": "Escribe un programa que implemente una tabla hash personalizada para almacenar y buscar claves enteras.", "salida\_esperada": "Valor encontrado: [valor]" },

{ "descripcion": "Escribe un programa que implemente el algoritmo de Dijkstra para encontrar la ruta más corta en un grafo.", "salida\_esperada": "Ruta más corta: [ruta]" },

{ "descripcion": "Escribe un programa que lea un archivo JSON y lo convierta en objetos Java usando una librería como Jackson o Gson.", "salida\_esperada": "[objetos deserializados]" },

{ "descripcion": "Escribe un programa que simule una tienda en línea con clases para productos, usuarios y compras.", "salida\_esperada": "[compra realizada]" },

{ "descripcion": "Escribe un programa que implemente un sistema de gestión de biblioteca con libros, préstamos y usuarios.", "salida\_esperada": "[préstamo registrado]" },

{ "descripcion": "Escribe un programa que implemente un motor de reglas básicas usando expresiones booleanas y reflexión.", "salida\_esperada": "Regla evaluada: [resultado]" },

{ "descripcion": "Escribe un programa que genere y resuelva un Sudoku de 9x9 con backtracking.", "salida\_esperada": "[solución del Sudoku]" }

]

[

{ "descripcion": "Escribe un programa que use reflexión para inspeccionar los métodos y atributos de una clase en tiempo de ejecución.", "salida\_esperada": "[lista de métodos y atributos]" },

{ "descripcion": "Escribe un programa que implemente el algoritmo A\* para encontrar la ruta óptima en un laberinto.", "salida\_esperada": "Ruta óptima: [ruta]" },

{ "descripcion": "Escribe un programa que utilice programación funcional con `Stream` para filtrar, mapear y reducir una colección de datos.", "salida\_esperada": "[resultado de la operación]" },

{ "descripcion": "Escribe un programa que cree un servidor web básico en Java usando sockets y maneje peticiones HTTP simples.", "salida\_esperada": "Respuesta HTTP: [respuesta]" },

{ "descripcion": "Escribe un programa que implemente un sistema de autenticación JWT (JSON Web Token) simulado.", "salida\_esperada": "Token generado: [token]" },

{ "descripcion": "Escribe un programa que conecte a una API REST pública, consuma datos y los visualice en consola.", "salida\_esperada": "[datos de la API]" },

{ "descripcion": "Escribe un programa que implemente un compilador muy básico para una mini-lengua (analizador léxico + sintáctico).", "salida\_esperada": "[resultado del análisis]" },

{ "descripcion": "Escribe un programa que simule un sistema bancario completo con múltiples cuentas, transferencias y registro de transacciones.", "salida\_esperada": "[transacción realizada]" },

{ "descripcion": "Escribe un programa que implemente un sistema de gestión de tareas con prioridades, fechas límite y persistencia en archivo.", "salida\_esperada": "[tarea guardada]" },

{ "descripcion": "Escribe un programa que use `CompletableFuture` para realizar múltiples tareas asincrónicas y combinar sus resultados.", "salida\_esperada": "[resultado combinado]" }

]

[

{ "descripcion": "Escribe un programa que implemente un sistema de recomendación utilizando algoritmos de filtrado colaborativo.", "salida\_esperada": "Recomendación: [recomendación]" },

{ "descripcion": "Escribe un programa que utilice inteligencia artificial para jugar al ajedrez, implementando un algoritmo minimax con poda alfa-beta.", "salida\_esperada": "Mejor movimiento: [movimiento]" },

{ "descripcion": "Escribe un programa que implemente un sistema de gestión de base de datos relacional, permitiendo crear tablas y ejecutar consultas SQL.", "salida\_esperada": "[resultado de la consulta]" },

{ "descripcion": "Escribe un programa que simule un sistema de procesamiento de imágenes, permitiendo aplicar filtros y transformaciones básicas.", "salida\_esperada": "[imagen procesada]" },

{ "descripcion": "Escribe un programa que utilice un algoritmo genético para resolver un problema de optimización.", "salida\_esperada": "Mejor solución: [solución]" },

{ "descripcion": "Escribe un programa que utilice Machine Learning para clasificar datos en dos categorías con una red neuronal simple.", "salida\_esperada": "Predicción: [categoría]" },

{ "descripcion": "Escribe un programa que implemente un servicio de microservicios en Java usando Spring Boot y conecte a una base de datos.", "salida\_esperada": "[respuesta del servicio]" },

{ "descripcion": "Escribe un programa que simule un sistema de votación en línea con autenticación, conteo de votos y resultados en tiempo real.", "salida\_esperada": "[resultados de la votación]" },

{ "descripcion": "Escribe un programa que implemente un sistema de búsqueda de texto utilizando índices invertidos y algoritmos de búsqueda eficientes.", "salida\_esperada": "[resultados de la búsqueda]" },

{ "descripcion": "Escribe un programa que implemente un modelo de red neuronal para reconocer patrones en imágenes utilizando una librería como TensorFlow.", "salida\_esperada": "Patrón reconocido: [patrón]" }

]