

### 1: Rotación de Arrays

Escribe un programa que permita al usuario ingresar un array de enteros y un número entero  $n$ . El programa debe rotar el array a la derecha  $n$  posiciones. Por ejemplo, si el array es  $[1, 2, 3, 4, 5]$  y  $n$  es 2, el resultado debe ser  $[4, 5, 1, 2, 3]$ .

### 2: Eliminación de Duplicados

Crea un programa que lea una lista de números enteros y elimine los elementos duplicados, manteniendo el orden original. Por ejemplo, si la entrada es  $[1, 2, 2, 3, 4, 4, 5]$ , la salida debe ser  $[1, 2, 3, 4, 5]$ .

### 3: Intersección de Arrays

Desarrolla un programa que tome dos arrays de enteros y encuentre los elementos comunes entre ambos, sin duplicados. Por ejemplo, para los arrays  $[1, 2, 3, 4]$  y  $[3, 4, 5, 6]$ , la salida debe ser  $[3, 4]$ .

### 4: Subarray con Suma Máxima

Implementa un programa que encuentre la suma máxima de un subarray contiguo dentro de un array de números enteros. Por ejemplo, para el array  $[-2, 1, -3, 4, -1, 2, 1, -5, 4]$ , la suma máxima sería 6, correspondiente al subarray  $[4, -1, 2, 1]$ .

### 5: Ordenamiento por Frecuencia

Escribe un programa que ordene los elementos de un array de enteros según la frecuencia de aparición de cada elemento, de mayor a menor frecuencia. Si dos elementos tienen la misma frecuencia, el menor número debe aparecer primero. Por ejemplo, para el array  $[4, 5, 6, 5, 4, 3]$ , la salida debe ser  $[4, 4, 5, 5, 3, 6]$ .

### 6: Combinación de Arrays Ordenados

Crea un programa que combine dos arrays ordenados de enteros en un solo array ordenado sin utilizar métodos de ordenamiento integrados. Por ejemplo, para los arrays  $[1, 3, 5]$  y  $[2, 4, 6]$ , la salida debe ser  $[1, 2, 3, 4, 5, 6]$ .

### 7: Encontrar el Elemento Faltante

Desarrolla un programa que encuentre el número faltante en un array que contiene  $n$  números únicos del 1 al  $n+1$ . Por ejemplo, para el array  $[1, 2, 4, 5, 6]$ , el número faltante es 3.

## 8: Rearreglar Array por Paridad

Implementa un programa que reorganice un array de enteros de manera que todos los números pares aparezcan antes que los impares, manteniendo el orden relativo de los números pares y de los impares. Por ejemplo, para el array `[3, 1, 2, 4]`, la salida debe ser `[2, 4, 3, 1]`.

## 9: Producto de Arrays Excepto el Índice

Escribe un programa que tome un array de enteros y devuelva un nuevo array tal que cada elemento en la posición `i` sea el producto de todos los elementos del array original excepto el elemento en `i`, sin utilizar la división. Por ejemplo, para el array `[1, 2, 3, 4]`, la salida debe ser `[24, 12, 8, 6]`.

## 10: Validación de Sudoku

Crea un programa que verifique si una matriz de 9x9 representa una solución válida de Sudoku. El programa debe comprobar que cada fila, cada columna y cada subcuadro de 3x3 contiene los números del 1 al 9 sin repetición.