

```

# interseccion de arreglos paralelos
"Ingrese numeros y cargar dos arreglos, generar nuevo arreglo con los comunes"

def cargar_datos(arreglo_1, arreglo_2):
    # 1. Ingreso de los tamaños N y M
    n = int(input("\nArreglo 1 cantidad de elementos: "))
    print("\nCargar el arreglo 1...\n")
    for i in range(n):
        arreglo_1.append(int(input(f"Ingrese el elemento {i+1}/{n} de N: ")))

    m = int(input("\nArreglo 2 cantidad de elementos: "))
    print("\nCargar el arreglo 2...\n")
    for i in range(m):
        arreglo_2.append(int(input(f"Ingrese el elemento {i+1}/{n} de M: ")))

def interseccion(arreglo_1, arreglo_2):
    arreglo_3 = []

    # Se recorre arreglo1 usando el índice 'i'
    for i in range(len(arreglo_1)):

        # Corrección: Uso directo del índice
        if arreglo_1[i] in arreglo_2 and arreglo_1[i] not in arreglo_3:
            arreglo_3.append(arreglo_1[i])

    return arreglo_3

def mostrar(arreglo):
    print(arreglo)

print("\n-----")

arreglo_a = []
arreglo_b = []
interseccion_c = []

cargar_datos(arreglo_a, arreglo_b)
print("\n-----")
print("\nElementos del arreglo 1: ")
mostrar(arreglo_a)
print("\nElementos del arreglo 2: ")
mostrar(arreglo_b)
if len(arreglo_a) > 0 and len(arreglo_b) > 0:
    print("\n-----")
    # AQUI ESTA LA CORRECCION CLAVE: ASIGNAR EL VALOR RETORNADO
    interseccion_c = interseccion(arreglo_a, arreglo_b)
    if len(interseccion_c) > 0:
        print("Los elementos en común son: ")
        mostrar(interseccion_c)
    else:
        print("No hay elementos en común.")
else:
    print("\nNo se ingresaron datos suficientes...")
print("\nFin del programa...")
print("\n-----\n")

```