

FUNCIONES	DESCRIPCIONES
generador_conj	sirve para crear un conjunto de dígitos únicos a partir de un número, en este caso un DNI Primero, inicializa una lista vacía llamada conjunto, luego recorre cada dígito del número, Por cada dígito, verifica si ya está en la lista. Si no está, lo agrega. De esta forma, al final la lista contiene solo los dígitos únicos del número, sin repetir ninguno. Finalmente, la función devuelve esa lista de dígitos únicos.
union	Realiza la unión de dos listas, combina los elementos de ambas listas sin repetir ninguno. Primero, crea una lista vacía llamada lista_u, recorre todos los elementos de la primera lista y los agrega a lista_u, solo si no existen, Después, hace lo mismo con la segunda lista: agrega cada elemento solo si no está en lista_u. De esta forma, al final, lista_u contiene todos los elementos de ambas listas, pero sin duplicados, representando la union de conjuntos matemática
inter	obtiene la intersección de dos listas, es decir, los elementos que están en ambas. Primero, crea una lista vacía llamada lista_i. Luego recorre la lista 2 comparando con los elementos de la primera lista y si es así la agrega a la lista i. al final retorna la lista que se repiten en ambas
dif	calcula la diferencia entre dos listas, es decir, obtiene los elementos que están en la primera lista pero no en la segunda. Primero, crea una lista vacía llamada lista_d. Luego, recorre cada elemento de la primera lista (lista1). Por cada elemento, verifica si NO está en la segunda lista (lista2). Si no está, lo agrega a lista_d. la función devuelve esa lista, que representa la diferencia de conjuntos en matemática:
difsim	Calcula la diferencia simétrica entre dos listas, es decir, obtiene los elementos que están en una lista o en la otra, pero no en ambas. Primero, crea una lista vacía llamada lista_ds. Luego, recorre cada elemento de la primera lista y, si ese elemento no está en la segunda, lo agrega a lista_ds. Después, hace lo mismo con la segunda lista: agrega los elementos que no están en la primera. Al final, la función devuelve la lista con todos los elementos que son exclusivos de cada lista, representando la diferencia simétrica de conjuntos en matemática.
rep_dig	cuenta cuántas veces aparece cada dígito en un número. verifica si el número es cero. Si es así, termina y devuelve el diccionario con los resultados. Si no, toma el último dígito del número usando el operador módulo (num % 10). Luego, revisa si ese dígito ya está en el diccionario cont. Si está, suma uno a su contador. Si no está, lo agrega con valor uno. Después, llama a la función nuevamente, pero con el número sin el último dígito (num // 10). Este proceso se repite hasta que el número llega a cero. Al final, la función devuelve un diccionario donde las claves son los dígitos y los valores son la cantidad de veces que aparece cada uno en el número. Esto es útil para analizar la frecuencia de los dígitos en un número.
conteo_dig	La función conteo_dig(num) sirve para contar cuántos dígitos tiene un número. Lo que hace es convertir el número a una cadena de texto usando str(num), y luego utiliza la función len() para contar cuántos caracteres tiene esa cadena.
diver	evalúa si una lista tiene al menos 6 elementos, lo que en este caso se interpreta como "diversidad numérica". inicializa un contador en cero. Luego, recorre la lista sumando uno al contador por cada elemento. Apenas el contador llega a 6, la función devuelve "Se cumple!", indicando que la lista tiene suficiente diversidad.
es_bisiesto	se usa para saber si un año es bisiesto. La condición dice que un año es bisiesto si se cumple una de dos cosas: Que sea divisible por 4 y no por 100, o Que sea divisible por 400.
ingresar_anios	Esta función sirve para pedirle al usuario que ingrese los años de nacimiento de los integrantes de un grupo. Primero, crea una lista vacía llamada anios. Luego, pregunta cuántos integrantes hay en el grupo y guarda ese número en la variable n. Después, utiliza un ciclo for para repetir la siguiente acción n veces: pide al usuario que ingrese el año de nacimiento de cada integrante y lo agrega a la lista anios. Al final, la función devuelve la lista con todos los años de nacimiento ingresados.
contar_pares_impares	sirve para contar cuántos años de una lista son pares y cuántos son impares. Primero, utiliza una expresión para sumar 1 por cada año que sea par (es decir, divisible por 2). Luego, calcula la cantidad de impares restando la cantidad de pares al total de años en la lista. Finalmente, la función devuelve dos valores: la cantidad de años pares y la cantidad de años impares.
todos_post_2000	sirve para verificar si todos los años de una lista son posteriores al año 2000.Utiliza la función all(), que devuelve True solo si todos los elementos cumplen una condición. En este caso, la condición es que cada año de la lista sea mayor a 2000. Si todos los años cumplen esto, la función devuelve True; si al menos uno no lo cumple, devuelve False.
alguno_bisiesto	sirve para saber si al menos uno de los años en una lista es bisiesto.Utiliza la función any(), que devuelve True si al menos un elemento cumple la condición.
calcular_edades	sirve para calcular la edad de cada persona. La función utiliza una lista por comprensión para restar cada año de nacimiento al año actual, obteniendo así la edad de cada persona.
producto_cartesiano	sirve para calcular el producto cartesiano entre dos listas: una de años y otra de edades.Lo que hace es combinar cada año con cada edad posible, formando pares ordenados. Utiliza una lista por comprensión con dos bucles: por cada año en la lista de años, recorre todas las edades y crea un par (anio, edad). El resultado es una lista con todos los pares posibles entre los elementos de ambas listas.
suma_digitos	sirve para sumar todos los dígitos de un número.Primeramente, convierte el número a una cadena de texto con str (num), así puede recorrer cada dígito por separado. Luego, para cada dígito, lo convierte de nuevo a entero y los va sumando todos usando la función sum().