*"El programa comienza mostrando los datos de los integrantes del equipo: nombres, DNIs y fechas de nacimiento. Introduce los datos que usaremos más adelante para las operaciones matemáticas."*

print("DATOS INTEGRANTES DEL GRUPO:")

**2. Operaciones con DNIs**  
*"La primera parte del programa se enfoca en los DNIs. Aquí realizamos tres acciones principales:"*

* **Ingreso de DNIs:**

. El código inicializa dos listas: nombres con los nombres de los integrantes y dnis con sus respectivos números de DNI. Luego, recorre estas listas y muestra cada nombre junto con su DNI asociado.

*Pausas Interactivas*

*"El programa incluye pausas con input() para que el usuario pueda leer los resultados paso a paso."*

**Generación de conjuntos de dígitos únicos:**  
*"Convertimos cada DNI en un conjunto de dígitos únicos, eliminando repeticiones. Por ejemplo, el DNI 17709420 se transforma en {1, 7, 0, 9, 4, 2}." Para esto utilizamos una lista*

2. Generación Automática de Conjuntos de Dígitos Únicos: En esta sección, el programa crea conjuntos de dígitos únicos para cada DNI. Para lograr esto, se inicializa una lista vacía llamada conjuntos que almacenará los conjuntos de dígitos. El código itera a través de cada DNI en la lista dnis:

Cada DNI, que inicialmente es un número entero, se convierte a una cadena de texto por medio de la función str y se guarda en dni\_texto para poder acceder a cada uno de sus dígitos individualmente.

Se crea un conjunto vacío (conjunto\_digitos) para cada DNI. Se utiliza la función set() que no permite que haya dígitos repetidos

Posteriormente, se recorre cada carácter (dígito) en la cadena dni\_texto. Cada dígito se convierte de nuevo a un número entero y se añade a conjunto\_digitos. Conjunto dígitos ahora tiene cada dni con dígitos enteros y sin repetir.

Finalmente, este conjunto\_digitos se añade a la lista conjuntos. El resultado es que se muestran los conjuntos de dígitos únicos para cada persona, ordenados para una mejor visualización (por ejemplo, "Conjunto A (Sergio Massazza): {1, 2, 4, 6, 7, 9, 0}"). El programa vuelve a pausarse, esperando que el usuario presione "Enter" para continuar.

3. Cálculo y Visualización de Operaciones entre Conjuntos: Esta es la parte central donde se aplican las operaciones matemáticas de conjuntos a los dígitos de los DNIs.

•

UNIÓN entre conjuntos: El código calcula y muestra la unión de cada par de conjuntos de dígitos. La unión de dos conjuntos (A ∪ B) incluye todos los elementos que están en A, o en B, o en ambos. Se utiliza un bucle anidado para asegurar que cada par de conjuntos se evalúe una sola vez (evitando repetir pares como A ∪ B y B ∪ A). La operación se realiza con conjuntos[i].union(conjuntos[j]).

•

INTERSECCIÓN entre conjuntos: Similar a la unión, se calcula y muestra la intersección de conjuntos. La intersección de dos conjuntos (A ∩ B) incluye solo los elementos que son comunes a ambos conjuntos. También se utiliza un bucle anidado para evitar la repetición de pares. La operación se realiza con conjuntos[i].intersection(conjuntos[j]).

•

DIFERENCIA entre pares de conjuntos: Finalmente, se calcula y muestra la diferencia entre pares de conjuntos. La diferencia de conjuntos A - B incluye los elementos que están en el conjunto A pero no en el conjunto B. El código calcula ambas diferencias para cada par: Conjunto A - Conjunto B y Conjunto B - Conjunto A, utilizando el operador - (por ejemplo, conjuntos[i] - conjuntos[j]).

En resumen, este código es un excelente ejemplo de cómo la programación (Python) puede usarse para visualizar y operar con conceptos matemáticos como la teoría de conjuntos, utilizando los dígitos de los DNIs como datos de entrada.