**Ejercicio 1:**

* + Define la función **main:**
    - Escriba un mensaje que diga **EJERCICIO 1**. Escriba un salto de línea con un **print()**.
    - Pide el número de renglones (arreglos a crear) y el número de columnas (número de elementos) de cada arreglo.
    - Genere una matriz de Numpy de puros **ceros** de tipo de dato "int" con el número de renglones y columnas solicitado. Llame a la matriz **m1**. Imprime la matriz.
    - Llama a la función **captura\_matriz** con la matriz creada **m1**.
    - Llama a la función **imprime\_matriz** con la matriz creada **m1**.
  + Define la función **captura\_matriz** que recibe una matriz. La función deberá pedir los valores para llenar la matriz siguiendo el orden que se muestra en el ejemplo.

**Casos de prueba:** Si la matriz es de 3 renglones x 2 columnas, entonces se pedirán 6 valores.



**Input:** Valor en [0, 0] es 0  
**Input:** Valor en [0, 1] es 1  
**Input:** Valor en [1, 0] es 2  
**Input:** Valor en [1, 1] es 3  
**Input:** Valor en [2, 0] es 4  
**Input:** Valor en [2, 1] es 5  
**Output:** La matriz creada con los valores

* + Define la función **imprime\_matriz (matriz)** que recibe una matriz. La función despliega en pantalla el contenido de la matriz.

**Ejemplo:** Suponiendo que la matriz es de 4 renglones x 4 columnas y tiene asignados los siguientes valores, la matriz se imprimiría de la siguiente forma:

 2  -5   6  -4  3   4   5  -1 -7   8   3   6  9  -7   2 10

    
**Ejercicio 2:**

* + Define la función **main:**
    - Escriba un mensaje que diga **EJERCICIO 2**. Escriba un salto de línea con un **print()**.
    - Pide el número de renglones (arreglos a crear) y el número de columnas (número de elementos) de cada arreglo.
    - Genere una matriz de Numpy de puros **unos** de tipo de dato "int" con el número de renglones y columnas solicitado. Llame a la matriz **m2**.
    - Imprima el atributo **shape** y el atributo **size** de la matriz **m2**.
    - Llama a la función **imprime\_matriz** con la matriz creada **m2**.
    - Llama a la función **llena\_matriz** con la matriz **m2**.
    - Llama a la función **imprime\_matriz** con la matriz **m2**.
    - Llama a la función **cuenta\_negativos** con la matriz **m2**. Imprime el resultado de la función.
  + Define la función **llena\_matriz (matriz)** que recibe una matriz. La función deberá modificar cada localidad de la matriz con valores aleatorios entre -25 y 20.  Importa el módulo random (**import random**). Utiliza la función **random.randint** para asignar un valor aleatorio a cada casilla de la matriz. **Ejemplo de cómo utilizar la función randint:** num = random.randint(10, 20) Genera un número aleatorio entre 10 y 20 incluyendo los límites.

**Ejemplo:** Suponiendo que la matriz es de 4 renglones x 4 columnas y se le asignaron los valores aleatorios.

 2  -25    6    -4  3     4    5    -1 -7     8    3     6  9    -7    2   20

* + Define la función **cuenta\_negativos (matriz)** que recibe una matriz de números enteros. La función regresará el resultado de contar todos los valores negativos contenidos en la matriz.

**Ejemplo:** Suponiendo que la matriz tiene asignados los siguientes valores:

 2  -5   6  -4  3   4   5  -1 -7   8   3   6  9  -7   2 10

La función regresaría: **5**

    
**Ejercicio 3:**

* + Define la función **main:**
    - Escriba un mensaje que diga **EJERCICIO 3**. Escriba un salto de línea con un **print()**.
    - Pide el número de **renglones** (arreglos a crear) y el número de **columnas** (número de elementos) de cada arreglo.
    - Genere una matriz de Numpy sin inicializar (**empty)** de tipo de dato "int" con el número de renglones y columnas solicitado. Llame a la matriz **mat1**.
    - Llama a la función **imprime\_matriz** con la matriz **mat1**.
    - Llama a la función **llena\_matriz** con la matriz creada **mat1**.
    - Imprime un salto de línea.
    - Escriba un mensaje que diga **MATRIZ 1**.
    - Llama a la función **imprime\_matriz** con la matriz **mat1**.
    - Imprime un salto de línea.
    - Genere una matriz con números aleatorios enteros entre 1 y 10 con el número de renglones y columnas solicitado. Llame a la matriz **mat2**.  Utiliza la función de numpy **random.randint**.
    - Imprime un salto de línea print().
    - Escriba un mensaje que diga **MATRIZ 2**.
    - Llama a la función **imprime\_matriz** con la matriz **mat2**.
    - Realice las siguientes operaciones con las matrices **mat1** y **mat2**:
      * **mat1 + mat2**
      * **mat1 - mat2**
      * **mat1 \* mat2**
      * **mat1 / mat2**
    - Guarde cada resultado en una variable llamada **mat3** e imprima la matriz resultante de cada operación con la función **imprime\_matriz**.
    - Concatena la matriz **mat1** y la matriz **mat2** una sobre otra, es decir, la matriz **mat1** arriba y la matriz **mat2** abajo. Guarda la nueva matriz en **mat4.**
    - Imprime un salto de línea print().
    - Escriba un mensaje que diga **MATRIZ 4**.
    - Llama a la función **imprime\_matriz** con la matriz **mat4**.

    
**Ejercicio 4:**

* + Define la función **main:**
    - Escriba un mensaje que diga **EJERCICIO 4**. Escriba un salto de línea con un **print()**.
    - Pide el número de **renglones** (arreglos a crear) y el número de **columnas** (número de elementos) de cada arreglo.
    - Genere una matriz con números aleatorios enteros entre 1 y 50 con el número de renglones y columnas solicitado. Llame a la matriz **mat**.  Utiliza la función de numpy **random.randint**.
    - Imprime un salto de línea.
    - Llama a la función **imprime\_matriz** con la matriz **mat**.
    - Pide el número de columna a modificar.
    - Llama a la función **multiplica\_columnax3** con la matriz creada y el número de columna solicitado
    - Imprime un salto de línea.
    - Llama a la función **imprime\_matriz** con la matriz **mat**.
    - Extrae la columna solicitada y guárdala en una variable llamada **arreglo**.
    - Imprime el arreglo.
    - Calcula el promedio del arreglo con la función de numpy **mean** e imprime el resultado.
  + Define la función **multiplica\_columnax3 (matriz, columna)** que recibe una matriz y el número de columna a modificar. La función deberá multiplicar toda la columna de la matriz por el valor de **3**.

**Ejemplo:** Suponiendo que la matriz tiene asignados los siguientes valores:

2  5  6  4 3  4  5  1 7  8  5  6 9  7  1  5

Al llamar **multiplica\_columna(matriz, 1)** los nuevos valores en la matriz deberán ser:

  2   **15**   6   4   3  **12**   5   1  7   **24**    5   6   9   **21**   1   5