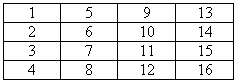
1. Desarrolla una función **crea\_matriz1** la cual recibe un entero n que representa el número de renglón y columnas. La función devuelve una lista que representa una matriz de nXn y cada localidad de la matriz tiene asignado el valor de -1.
2. Desarrolla una función **matriz\_columna** la cual recibe un entero n que representa el número de renglón y columnas. La función devuelve una lista que representa una matriz de nXn y cada localidad de la matriz tiene asignado el número de su respectiva columna.
3. Desarrolla una función **matriz\_renglon** la cual recibe un entero n que representa el número de renglón y columnas. La función devuelve una lista que representa una matriz de nXn y cada localidad de la matriz tiene asignado el valor de su correspondiente renglón.
4. Escribe la función **matriz\_aleatoria**, dos enteros, que representan la cantidad de renglones y columnas que debe tener la matriz y regresa una matriz a la cual se le asignaron números aleatorios entre 0 y 100.
5. Desarrolla una función **matriz\_secuencia** la cual recibe un entero n que representa el número de renglón y columnas. La función devuelve una lista que representa una matriz de nXn y cada localidad de la matriz tiene asignado un número consecutivo a partir de uno, ***por renglones***. Observa el ejemplo:

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

1. Desarrolla una función **matriz\_secuencia\_columna** la cual recibe un entero n que representa el número de renglón y columnas. La función devuelve una lista que representa una matriz de nXn y cada localidad de la matriz tiene asignado un número consecutivo a partir de uno, ***por columnas***. Observa el ejemplo:



1. Desarrolla una función **cuenta\_pares**, la cual recibe como parámetro una lista anidada de números y regresa cuántos números pares hay dentro de la lista.
2. Desarrolla una función **cuenta\_positivos**, que reciba como parámetro una lista anidada de números. La función regresa el número de valores mayores o iguales que cero almacenados en la lista.
3. Desarrolla una función **cambia\_negativos**, que reciba como parámetro una lista anidada de números. La función modifica la lista, asignando el valor de 0 a cada localidad que contenga un valor menor a cero.
4. Desarrolla una función **cuenta\_repeticiones** que recibe como parámetro una lista anidada de números y un valor entero x. La función regresa el número de veces que se repite el valor de x en la lista.
5. Desarrolla una función **busca** que recibe como parámetro una lista anidada de números y un valor entero x. La función regresa el valor **True** si el valor de x existe en una localidad de la matriz y **False** si no existe.
6. Desarrolla una función **suma\_mayores5** que recibe como parámetro una lista anidada de números. La función deberá devolver la suma de todos aquellos números que haya dentro de la matriz que sean mayores a 5.
7. Desarrolla una función **cambia\_ceros** que recibe como parámetro una lista anidada de números con muchos ceros dentro de ella. La función deberá sustituir todos aquellos ceros que haya dentro de la matriz por el resultado de la suma de su número de renglón por su número de columna.

**RÚBRICA DE EVALUACIÓN**

|  |  |
| --- | --- |
| Concepto a evaluar | Puntaje |
| Declaración de todas las funciones al inicio del programa. | 10 |
| Formato en tu programa para separar las funciones una de otra | 10 |
| Documentación de la función con *docstrings*. | 10 |
| Ciclo y condicionales anidados para hacer un menú principal y todas las funciones son llamadas como opciones del programa cuidando qué funciones deben llamarse consecutivamente. | 20 |
| Todas las funciones corriendo correctamente mostrando los resultados esperados con uso de for anidado para el llenado, impresión o búsqueda en la lista de listas (matriz) | 30 |
| Experiencia del usuario: El Menú es claro. La captura de datos es entendible por el usuario y se despliegan los resultados con mensajes correctos. Se entiende qué hace cada programa y qué se le está solicitando al usuario.  *La presentación del programa es agradable a la vista, se hace un BUEN manejo de la pantalla, se utilizan formatos para las impresiones y el acomodo visual es atractivo* | 20 |
|  |  |