**ARREGLOS**

* + Genere un arreglo de Numpy de puros unos de tipo de dato "int" de 50 elementos. Imprime el arreglo.
  + Genere un arreglo de Numpy con 20 elementos de tipo float sin inicializar. Usa la función **empty**. Imprime el arreglo.
  + Genere un arreglo de Numpy con números que vayan del 5 al 12. Usa la función **arange()**. Imprime el arreglo.
  + Genere un arreglo de Numpy con números entre 1 y 22 con incrementos de 3, es decir: [1 4 7 10 13 16 19 22]. Usa la función **arange()**. Imprime el arreglo.
  + Genere un arreglo de Numpy con 15 elementos de tipo float entre 20 y 40 igualmente espaciados. Usa la función **linspace**. Imprime el arreglo.
  + Genera un arreglo de Numpy llamado **arreglo1** con 5 números aleatorios enteros entre 10 y 20  y genera otro arreglo llamado **arreglo2** con 7 números aleatorios enteros entre 50 y 60. Concatena ambos arreglos en un arreglo llamado **arreglo3**. Imprime los tres arreglos.
  + Crea la función main():
    - Genera un arreglo de Numpy con 10 números aleatorios enteros entre -5 y 15. Imprime el arreglo.
    - Crea la función **arreglo\_positivo** que recibe el arreglo anterior. La función deberá hacer positivas todas aquellas localidades que tengan un valor negativo. Regresa el arreglo modificado.
    - En el main llama a la función **arreglo\_positivo** e imprime el arreglo modificado.

**Ejemplo:** Si el arreglo original tiene asignados los siguientes valores:

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **-1** | **3** | **5** | **10** | **-3** | **12** | **-5** |
| **0** | **1** | **2** | **3** | **4** | **5** | **6** |

   Al aplicar la función **arreglo\_positivo (arreglo)** el arreglo quedaría con los siguientes valores:

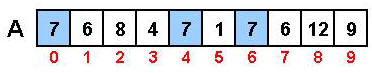
|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| **1** | **3** | **5** | **10** | **3** | **12** | **5** |
| **0** | **1** | **2** | **3** | **4** | **5** | **6** |

* Crea la función **main()**:
  + Genera un arreglo de Numpy con 10 números aleatorios enteros entre 1 y 12. Imprime el arreglo.
  + Crea la función **sustituye** que recibe el **arreglo**, un valor **x** y un valor **y**. La función deberá sustituir las ocurrencias de **x** en el arreglo por el valor de **y**. Regresa el arreglo modificado.
  + En el main pide el valor de **x** y **y**, llama a la función **sustituye** e imprime el arreglo modificado.

**Ejemplo:** Si el arreglo tiene asignados los siguientes valores:



        Al aplicar la función **sustituye(arreglo, 3, 7)**, suponiendo que x = 3, y = 7, el arreglo quedaría con los siguientes valores:



* Crea la función **main()**:
  + Crea un arreglo en Numpy con los siguientes valores: [1, 3, 5, 10, -3, 12, 5, 9, 9, -5].
  + Crea la función **suma\_multiplos** que recibe el **arreglo** y un valor **x**. La función deberá regresar el resultado de sumar todos los elementos múltplos de **x** contenidos en el arreglo. Regresa el resultado.
  + En el main pide el valor de **x** y llama a la función **suma\_multiplos** e imprime el resultado.

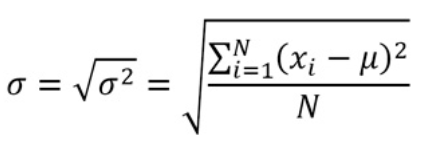
**Ejemplo:**



                    Al aplicar la función **suma\_multiplos (arreglo, x)** y suponiendo que **x** tuviera el valor de **5**. El valor que regresaría la función sería: **15**.

* + Crea la función **main()**:
    - Genere un arreglo de Numpy con los siguientes números flotantes: [2.8, 8.3, 5.4, 6.0, 5.2, 1.5, 6.4, 12, 7.1, 3.6].
    - Utiliza la función **std** de numpy y calcula la desviación estándar del arreglo anterior. Imprime el resultado.
    - Crea la función **desviacion** que recibe el arreglo anterior y con un ciclo iterador calcula la desviación estándar. Regresa el resultado. Nota: Puedes usar la función **mean** para calcular el promedio del arreglo.
    - En el main llama a la función **desviación** e imprime el resultado. Revisa que tu resultado concuerde con el resultado anterior.

La fórmula para calcular la desviación estándar es la siguiente:



**Ejemplo:** Suponiendo que el arreglo tuviera asignados los siguientes valores:

|  |  |  |  |
| --- | --- | --- | --- |
| **2** | **4** | **6** | **8** |
| **0** | **1** | **2** | **3** |

El resultado de la función sería: **2.236**

