**Ejercicios Complementarios**

✨ Estos ejercicios **son de tipo complementario**. Esto quiere decir que te ayudará a avanzar en profundidad en el tema visto, pero **no son obligatorios**. Te recomendamos intentar con tu equipo trabajar al menos uno de ellos.

1. Desarrolla un programa que ordene un vector lleno de números enteros aleatorios de menor a mayor. La dimensión del vector debe ser solicitada al usuario. Puedes investigar el método de ordenamiento burbuja para implementarlo. Para obtener más información sobre el ordenamiento burbuja, puedes consultar el siguiente enlace:
   1. [¿Qué es el ordenamiento burbuja?](https://www.studocu.com/row/document/sveuciliste-u-zagrebu/fiziologija/metodos-de-ordenacion-de-vectores-en-pse-int/17112051)
   2. 🎥 [Ordenamiento Burbuja- Pseint](https://youtu.be/Xa5nH03RGOI?feature=shared)
2. Desarrolla una función recursiva que calcula la suma de un arreglo de números enteros o reales, según la preferencia del usuario. La función debe tomar como entrada el arreglo, su longitud y el tipo de números contenidos en él. Luego, devolverá la suma total de todos los elementos del arreglo. Esta función puede ser aplicada a arreglos de cualquier tamaño deseado, permitiendo al usuario manipular conjuntos de datos de manera eficiente y versátil. La implementación de esta función proporcionará práctica en el uso de la recursión para operaciones sobre arreglos.
3. Desarrolla un programa que inicialice un arreglo de números, permitiéndote elegir el tipo y tamaño del arreglo. Asigna valores aleatorios entre 15 y 155. Posteriormente:
4. Crea una función para calcular la suma de TODOS LOS ELEMENTOS CON ÍNDICE IMPAR. El número 0(cero) no será contemplado en este grupo. Es decir, los elementos del vector con índice 1,3,5 etc.
5. Crea una función para calcular la suma de TODOS LOS ELEMENTOS CON ÍNDICE PAR. El número 0(cero) debe ser contemplado en este grupo. Es decir, los elementos del vector con índice 0,2,4,6 etc.
6. Diseña un programa que trabaje con un vector de cadenas de caracteres de dimensión 10, donde cada cadena representa un nombre ingresado por el usuario. El programa debe realizar las siguientes tareas:
7. Llenar el vector con nombres ingresados por el usuario.
8. Mostrar el vector original en pantalla.
9. Contar y mostrar cuántas veces se repite cada vocal en todos los nombres del vector. Es decir el total de A-E-I-O-U utilizadas en todo el vector.
10. Encontrar y mostrar el nombre más largo y el nombre más corto del vector.
11. Crear un nuevo vector que contenga solo los nombres que comiencen con una letra específica ingresada por el usuario.Mostrar el nuevo vector en pantalla.