

## Fundamentos de programación.

### Tarea sobre Listas.

Para esta tarea puedes utilizar **Calico** o **PyCharm** para resolverla. Al terminarla solo debes subir el archivo **Tarea08\_tuPrimerNombre.py** a Github.

1. Escribe una función que recibe como parámetro una lista de números enteros y regresa UNA NUEVA lista con la suma acumulada de los datos.  
Esta nueva lista tiene en la posición  $i$ , la suma de los primeros  $i+1$  valores de la lista original. Por ejemplo, si recibe la lista `[1, 2, 3]`, regresa la lista `[1, 3, 6]`.
2. Escribe una función que recibe como parámetro una lista y regresa una nueva lista, pero sin el primero y último elemento.  
Por ejemplo, si recibe `[1,2,3,4,5]`, regresa `[2,3,4]`; si recibe `[1,2]`, regresa `[]`. Si recibe `[]`, regresa `[]`.
3. Escribe una función que recibe una lista de valores y regresa *True* si los valores están ordenados, *False* en otro caso. Por ejemplo, si recibe `[1,2,3]` regresa *True*, si recibe `["A", "X", "B"]` regresa *False*.
4. Escribe una función que recibe dos cadenas, regresa *True* si son anagramas, *False* en otro caso. Por ejemplo, si recibe `"Roma"` y `"Mora"` regresa *True*, si recibe `"Hola"` y `"Hello"` regresa *False*.
5. Escribe una función que recibe una lista y regresa *True* si alguno de sus datos está duplicado, *False* si todos son únicos. Por ejemplo, si recibe `[1,2,3,1,4,5]` regresa *True*, si recibe `[5,4,3,2,1]` regresa *False*.
6. Escribe una función que recibe una lista de valores y elimina de ésta los valores repetidos (solo deja uno). Por ejemplo, si recibe `[1,8,3,4,3,1,2,7]` la modifica a `[1,8,3,4,2,7]`. No importa el orden. NO REGRESA una nueva lista, hace los cambios sobre la misma.
7. Finalmente, escribe una función **main** que pruebe cada una de las funciones con varios casos para demostrar que funciona correctamente.  
Por ejemplo, para la función 1, puedes escribir varias pruebas:
  - La lista `[1,2,3,4,5]` regresa la lista acumulada `[1,3,6,10,15]`
  - La lista `[]` regresa la lista acumulada `[]`
  - La lista `[5]` regresa la lista acumulada `[5]`