Suponiendo que cuenta con una lista de nombres:

1. Escribe una función que reciba dos palabras y sustituya la primera por la segunda en la lista.
2. Escribe una función que una palabra y elimine todas las apariciones de esta palabra en una lista.
3. Escribe una función que reciba por parámetro una lista listaB y que elimine de la lista original los nombres de la segunda lista.
4. Escribe una función que reciba por parámetro una lista listaB y con base en la lista original cree una tercera lista donde incluya las palabras que están en las dos listas. (sin repeticiones).
5. Escribe una función que reciba por parámetro una lista listaB y con base en la lista original cree una tercera lista donde incluya las palabras que están en la lista original pero no en la segunda.
6. Escribe una función que reciba por parámetro una lista listaB y con base en la lista original cree una tercera lista donde incluya las palabras que están en la listaB pero no en la original.
7. Escribe una función que reciba por parámetro una lista listaB y con base en la lista original cree una tercera lista donde incluya las palabras que están en ambas listas.
8. Escribe una función que cree una segunda lista con las palabras de la original, pero sin palabras repetidas (el orden de las palabras en la segunda lista no es importante).
9. Escribe una función que cree una lista con los n términos de una sucesión del tipo Un+1 = a Un + b. La función recibe el valor de a, de b y del término U0 y el número de términos a calcular.

Otros ejercicios sobre listas:

1. Escribe una función que cree una lista con los n términos de la sucesión Un+1 = 3 Un + 1 si Un es impar y U n+1 = Un / 2 si Un es par. La función recibe el valor del término U0 y el número de términos a calcular.
2. Cuando en un número la diferencia entre cada par de dígitos consecutivos es uno, se lo llama número "step" (como el 123234, el 9876787654, etc.). Escriba una función que se llame **averiguaSteps**() que muestre la lista de números step que hay entre 1 y 100.000.
3. Escriba una función llamada **igualCantDiv** que muestre en una lista los enteros entre 1 < n < 107 para los cuales n y n+1 tienen la misma cantidad de divisores positivos enteros. Por ejemplo, 14 tiene el 1, 2, 7 y 14, mientras que 15 tiene 1, 3, 5 y 15 y entonces 14 y 15 forman parte de la lista.