



Guía de ejercicios de la primera clase

Ejercicios

1. Definir una función `pertenece(lista, elem)` que tome una lista y un elemento, y devuelva `True` si la lista tiene al elemento dado y `False` en caso contrario.
2. Definir una función `mas_larga(lista1, lista2)` que tome dos listas y devuelva la más larga.
3. Definir la función `mezclar(cadena1, cadena2)` que tome dos strings y devuelva el resultado de intercalar elemento a elemento. Por ejemplo: si intercalamos Pepe con Jose daría PJeopsee. En el caso de Pepe con Josefa daría PJeopseefa.
4. David solicitó un crédito a 30 años para comprar una vivienda, con una tasa fija nominal anual del 5%. Pidió \$500000 al banco y acordó un pago mensual fijo de \$2684,11.
 - a. Escribir un programa que calcula el monto total que pagará David a lo largo de los años. Deberías obtener que en total paga \$966279.6.
 - b. Supongamos que David adelanta pagos extra de \$1000/mes durante los primeros 12 meses de la hipoteca. Modificá el programa para incorporar estos pagos extra y que imprima el monto total pagado junto con la cantidad de meses requeridos. Deberías obtener que el pago total es de \$929965.62 en 342 meses.
 - c. ¿Cuánto pagaría David si agrega \$1000 por mes durante cuatro años, comenzando en el sexto año de la hipoteca (es decir, luego de 5 años)? Modificá tu programa de forma que la información sobre pagos extras sea incorporada de manera versátil. Sugerimos utilizar los parámetros: `pago_extra_monto`, `pago_extra_mes_comienzo`, `pago_extra_mes_fin`.
5. Construí una función `traductor_geringoso(lista)` que, a partir de una lista de palabras, devuelva un diccionario geringoso. Las claves del diccionario deben ser las palabras de la lista y los valores deben ser sus traducciones al geringoso. Por ejemplo, al tomar la lista `['banana', 'manzana', 'mandarina']` debe devolver `{'banana': 'bapanapanapa', 'manzana': 'mapanzapanapa', 'mandarina': 'mapandaparipinapa'}`.



6. Escribí un programa que imprima de forma prolija las tablas de multiplicar del 1 al 9 usando f-strings. Si podés, evitá usar la multiplicación, usando sólo sumas alcanza.

	0	1	2	3	4	5	6	7	8	9
0:	0	0	0	0	0	0	0	0	0	0
1:	0	1	2	3	4	5	6	7	8	9
2:	0	2	4	6	8	10	12	14	16	18
3:	0	3	6	9	12	15	18	21	24	27
4:	0	4	8	12	16	20	24	28	32	36
5:	0	5	10	15	20	25	30	35	40	45
6:	0	6	12	18	24	30	36	42	48	54
7:	0	7	14	21	28	35	42	49	56	63
8:	0	8	16	24	32	40	48	56	64	72
9:	0	9	18	27	36	45	54	63	72	81