

# Trabajo Práctico 2: Listas en Python.

## Sección 1: Operaciones básicas de listas

1. Construcción de listas:
  - a. Construir una lista que contenga los números pares del 0 al 20 usando `append()` en un bucle `for`.
  - b. Construir la misma lista de a. pero en un bucle `while`
  - c. Mediante listas por comprensión
  - d. Usando la función `list()` y un iterable
2. Escriba un programa que solicite al usuario un número entero positivo `n` y luego utilice solicite al usuario el ingreso de `n` datos de tipo `Float` y colóquelos en una lista.
3. Escriba un programa que haga lo mismo que el pedido en 2. pero para el caso de una matriz 2D (una lista de listas de enteros).

**Nota:** Una lista de lista se puede asignar de la siguiente forma `matriz = [[1, 2, 3], [4, 5, 6], [7, 8, 9]]` o posición a posición haciendo `matriz[i][j] = valor`.

## Sección 2: Resolución de problemas sobre listas

Use el programa del ejercicio e. de la sección anterior como primer paso en la resolución de los siguientes problemas.

4. Muestre por pantalla las posiciones pares de una lista y luego las impares
5. Muestre por pantalla una lista en orden inverso
6. Busque el mínimo de una lista (hacerlo tanto con la función `min` como a través de un ciclo y al final compare que estos números sean iguales)
7. Busque el máximo de una lista (hacerlo tanto con la función `max` como a través de un ciclo y al final compare que estos números sean iguales)
8. Solicite al usuario el ingreso de 2 datos de tipo `Float` `[n, m]` y muestre aquellos de elementos de una lista que son mayores que `n` y menores que `m`.
9. Escriba un programa que coloca el valor `True` en una variable de tipo `bool` si una lista está ordenada y `False` en caso contrario. Al terminar muestre el valor de esta variable.
10. **(difícil)** Ordene la lista (hacerlo tanto con la función `sort` como a través de un programa que utilice la función `copy` que copia una lista, `min` y `remove` que elimina un elemento de una lista)
11. Escriba un programa que dadas dos listas ordenadas las mezcle en una única lista ordenada, preservando el orden relativo de las primeras.

**Ejemplo:** entrada: `[1,5,9,12]`, `[2,4,7,10]` - salida: `[1,2,4,5,7,9,10,12]`
12. Escriba un programa que compute el índice de comienzo y final de la meseta más larga de una lista de números. Una meseta es una sublista de números iguales.

**Ejemplo:** entrada: `[1,3,3,1,1,7,7,7,7,4,5,5,4,4,4]` - salida: `[5, 9]` (recordar que siempre consideramos los rangos con el límite superior no inclusive)

## Sección 3: Listas por comprensión

13. Crea una lista con los cuadrados de los números del 0 al 9
14. Genera una lista con los números pares del 0 al 19.
15. Dada una lista de palabras, crea una nueva lista con las palabras en mayúsculas.
16. Encuentra los números entre 1 y 100 que son divisibles por 3 y 5 simultáneamente.
17. Convierte una matriz 2D (una lista de listas de enteros) en una lista 1D de enteros.
18. Dada una lista con tipos variados (int, float y bool), extrae solo los valores numéricos (enteros o flotantes).  
**Nota:** Utilice la función `isinstance([variable], [tipo])`, por ejemplo, `isinstance(x, int)` o `isinstance(x, float)`
19. Genera una lista de números primos entre 2 y 50.