

```

#Segundo Parcial Modelo 2 del jueves
"""

Pregunta 2
Ingresar numeros positivos hasta cargar un arreglo con tres numeros pares múltiplos de 7
y 3 numeros impares múltiplos de 3 en ese orden.
mostrar el arreglo ejemplo se ingresan 21 14 2 15 28 31 7 42 27
el arreglo queda: 14 28 42 21 15 27
Ingresa números positivos hasta cargar un arreglo con
1. Tres números pares múltiplos de 7.
2. Tres números impares múltiplos de 3.
3. Muestra el arreglo resultante.
"""

#FUNCIONES
def cargar_datos(arr_par7, arr_mult3, arreglo):

    cont_arr_par_7 = 0
    cont_arr_mult3 = 0

    # El ciclo sigue mientras falten datos en alguna de las dos listas
    while len(arr_par7) < 3 or len(arr_mult3) < 3:

        numeros = int(input("\nIngrese un número: "))

        # 1. Verificar si es PAR, Múltiplo de 7 Y si hay espacio en la lista de pares
        if numeros % 2 == 0 and numeros % 7 == 0 and len(arr_par7) < 3:
            arr_par7.append(numeros)
            cont_arr_par_7 += 1
            print(f"Pares Múltiplos de 7 cargados: {cont_arr_par_7}/3")

        # 2. Verificar si es IMPAR, Múltiplo de 3 Y si hay espacio en la lista de impares
        # Usamos ELIF para que un número no pueda entrar en ambas listas (aunque
        # matemáticamente son excluyentes aquí)
        elif numeros % 2 != 0 and numeros % 3 == 0 and len(arr_mult3) < 3:
            arr_mult3.append(numeros)
            cont_arr_mult3 += 1
            print(f"IMPARES Múltiplos de 3 cargados: {cont_arr_mult3}/3")

        else:
            print(f"{numeros} ignorado (no cumple condición o lista llena).")

def mostrar_datos(arr_par7, arr_mult3, arreglo):
    # Se concatena al final para mostrar primero los del grupo 7 y luego los del grupo 3
    arreglo_final = arr_par7 + arr_mult3
    print(f"El arreglo es: {arreglo_final}")

# PPAL
print("\n-----")
arreglo = []
pares7 = []
multiplos3 = []

print("\nCargando arreglo con 3 pares múltiplos de 7.")
print("Cargando arreglo con 3 impares múltiplos de 3.")

cargar_datos(pares7, multiplos3, arreglo)

```

```
if len(pares7) > 0 or len(multiplos3) > 0:  
    print("\n-----")  
    mostrar_datos(pares7, multiplos3, arreglo)  
else:  
    print("\n-----")  
    print("No se pudieron cargar datos.")  
print("\n-----")
```