

""""Se requiere realizar la solución al siguiente problema: Lucas y Micaela, poseen una agencia de venta de autos, y solicitan a ud. la realización de un programa informático. Puntualmente el sistema que le solicitan es de registro de pedidos. Este registro se realiza al finalizar el día. Se cargan los datos de las solicitudes HASTA QUE el modelo sea "FIN". Los modelos son Familiar y Utilitario, una vez cargado el modelo se carga la cantidad solicitada. (dos vectores) Se muestran en formato tabla, los datos ingresados:

- 1.- Se calcula la cantidad mínima solicitada y cual es el modelo asociado (SIN ORDENAR EL VECTOR).
- 2.- Calcular y mostrar los porcentajes pedidos para ambos modelos (según las cantidades solicitadas) sobre el total
- 3.- Ordenar los vectores según las cantidades solicitadas de forma programática (TENGA EN CUENTA QUE AL TRATARSE DE DOS VECTORES, EL ORDENAMIENTO TIENE QUE ESTAR RELACIONADO)
- 4.- Indicar cual fue el promedio de cantidades solicitadas por cada modelo
- 5.- Donde se encuentre una solicitud de un vehículo Familiar con la cantidad múltiplo de 3, se reemplazarán los valores por "reemplazo" y -1
- 6.- Eliminar los datos de ambos vectores donde haya una cantidad solicitada de 7 unidades. mostrar todo lo solicitado"""

----- FUNCIONES -----

```
def cargar_pedidos(arr_mod, arr_cant):
    print("--- INGRESO DE PEDIDOS ---")
    # 1. Lectura inicial
    modelo = input("\nIngrese Modelo F/U (FIN para terminar): ").upper()
    # 2. El ciclo principal solo se rompe con "FIN"
    while modelo != "FIN":
        # 3. Validamos que el modelo sea correcto SI no es "FIN"
        while modelo != "F" and modelo != "U" and modelo != "FIN":
            modelo = input("Error - Ingrese Modelo válido (F/U) o FIN: ").upper()
        # 4. Verificamos nuevamente si el usuario puso FIN tras el error para no pedir
        cantidad
        if modelo != "FIN":
            cantidad = int(input(f"Cantidad solicitada para {modelo}: "))
            while cantidad < 0:
                cantidad = int(input(f"Error (Mínimo 0) - Cantidad para {modelo}: "))
            arr_mod.append(modelo)
            arr_cant.append(cantidad)

            modelo = input("\nIngrese Modelo F/U (FIN para terminar): ").upper()
```

```
def mostrar_tabla(arr_mod, arr_cant):
    # Muestra los datos en formato tabla.
    titulo = "Modelo"
    titulo1 = "Cantidad"
    print(f"\n{titulo:<15}{titulo1}")
    print("-"*20)
    i = 0
    while i < len(arr_mod):
        print(f"{arr_mod[i]:<15}{arr_cant[i]}")
        i += 1
```

1.- Se calcula la cantidad mínima solicitada.

```
def buscar_minimo(arr_cant):
    # Encuentra la posición del valor mínimo.
    pos_min = 0
```

```

for i in range(1, len(arr_cant)):
    if arr_cant[i] < arr_cant[pos_min]:
        pos_min = i
    i += 1
return pos_min

def suma_cant(arr_cant):
    # Calcula la suma de todas las cantidades.
    acum = 0
    i = 0
    while i < len(arr_cant):
        acum += arr_cant[i]
        i += 1
    return acum

# 2.- Calcular y mostrar los porcentajes pedidos para ambos modelos cantidades
solicitadas/el total.
def calcular_porcentajes(total_cant,total_fam,total_util):
    if total_cant > 0:
        por_fam = (total_fam / total_cant) * 100
        por_util = (total_util / total_cant) * 100

        print(f"\nPorcentaje Familiar: {por_fam:.2f}%")
        print(f"Porcentaje Utilitario: {por_util:.2f}%")

def intercambiar(arr_mod, i, j):
    # Intercambia dos elementos en un arreglo (para ordenamiento).
    aux = arr_mod[i]
    arr_mod[i] = arr_mod[j]
    arr_mod[j] = aux

# 3.- Ordenar los vectores según las cantidades solicitadas de forma programática
(RELACIONADO)
def ordenar(arr_mod, arr_cant):
    # Ordena los vectores de forma ascendente por cantidad.
    for i in range(len(arr_cant)- 1):
        for j in range(i + 1, len(arr_cant)):
            if arr_cant[i] > arr_cant[j]:
                # Intercambiamos ambos vectores para mantener el paralelismo
                intercambiar(arr_cant, i, j)
                intercambiar(arr_mod, i, j)

def promedio(arr_importes):
    # Calcula el promedio de un arreglo de números.
    acum = 0.0
    cantidad_elementos = 0
    i = 0
    while i < len(arr_importes):
        acum += arr_importes[i]
        cantidad_elementos += 1
        i += 1
    if cantidad_elementos > 0:
        prom_total = acum / cantidad_elementos
    return prom_total

# 4.- Indicar cual fue el promedio de cantidades solicitadas por cada modelo

```

```

def prom_fami(arr_mod,arr_cant):
    # Calcula promedio de modelo familiar.
    total_modelos = suma_cant(arr_cant)
    familiar = 0
    cont_fami = 0
    i = 0
    while i < len(arr_mod):
        if arr_mod[i] == "F":
            familiar += arr_cant[i]
            cont_fami += 1
        i += 1
    if cont_fami > 0:
        prom_familiar = (cont_fami/total_modelos)
    return prom_familiar

# Calcula promedio de modelo utilitario.
def prom_util(arr_mod,arr_cant):
    total_modelos = suma_cant(arr_cant)
    utilitario = 0
    cont_util = 0
    i = 0
    while i < len(arr_mod):
        if arr_mod[i] == "U":
            utilitario += arr_cant[i]
            cont_util += 1
        i += 1
    if cont_util > 0:
        prom_utilitario = (cont_util/total_modelos)
    return prom_utilitario

# 5.- Reemplazar Familiar con la cantidad múltiplo de 3 por "reemplazo" y -1
def reemplazar_familiar_mult3(arr_mod, arr_cant):
    i = 0
    while i < len(arr_mod):
        if arr_mod[i] == "F":
            if arr_cant[i] % 3 == 0:
                arr_mod[i] = "Reemplazo"
                arr_cant[i] = -1
            i += 1

# 6.- Eliminar los datos donde la cantidad solicitada sea 7 unidades.
def eliminar_cantidad_7(arr_mod, arr_cant):
    i = 0
    while i < len(arr_cant):
        if arr_cant[i] == 7:
            arr_mod.pop(i)
            arr_cant.pop(i)
        else:
            i += 1

# ----- PROGRAMA PRINCIPAL -----
print("\n-----\n")
modelos = []
cantidades = []

cargar_pedidos(modelos, cantidades)
print("\n-----")

```

```

if len(cantidades) > 0:
    # Mostrar tabla inicial
    print("\nDatos Ingresados:")
    mostrar_tabla(modelos, cantidades)
    print("\n-----")

    # 1. Mínimo solicitado
    index_min = buscar_minimo(cantidades)
    print(f"\nMínimo solicitado: {cantidades[index_min]} ({modelos[index_min]})")
    print("\n-----")

    # 2. Porcentajes
    total_gral = suma_cant(cantidades)
    tot_fam = suma_cant(cantidades)
    tot_util = suma_cant(cantidades)
    calcular_porcentajes(total_gral, tot_fam, tot_util)
    print("\n-----")

    # 3. Ordenamiento
    ordenar(modelos, cantidades)
    print("\nDatos ordenados por cantidad (ascendente):")
    mostrar_tabla(modelos, cantidades)
    print("\n-----")

    # 4. Promedios por modelo
    promedio_util = prom_util(modelos, cantidades)
    promedio_fami = prom_util(modelos, cantidades)
    if promedio_util > 0:
        print(f"\nPromedio Familiar: {(promedio_fami):.2f}")
    else:
        print("Promedio Familiar: No hay datos.")

    if promedio_util > 0:
        print(f"\nPromedio Utilitario: {(promedio_util):.2f}")
    else:
        print("Promedio Utilitario: No hay datos.")
    print("\n-----")

    # 5. Reemplazo
    reemplazar_familiar_mult3(modelos, cantidades)
    print("\nDespués de reemplazar modelo F múltiplos de 3:")
    mostrar_tabla(modelos, cantidades)
    print("\n-----")

    # 6. Eliminación
    eliminar_cantidad_7(modelos, cantidades)
    print("\nDespués de eliminar modelos con cantidad 7:")
    if len(cantidades) > 0:
        mostrar_tabla(modelos, cantidades)
    else:
        print("\nLa lista quedó vacía tras las eliminaciones.\n")
    print("\n-----")

    # Ejemplo de uso de la función 'promedio' general.
    promedioCantidades = promedio(cantidades)
    print(f"El promedio general de las cantidades solicitadas es:

```

```
{promedioCantidades:.2f}")
else:
    print("No se ingresaron pedidos.")
print("-----\n")
print("Fin del programa...")
print("-----\n")
```