```python
#Segundo Parcial Nancy Guerrero
"""desarrollar un programa que cumpla con las siguientes funciones:
Requisitos del programa:
-Ingreso y almacenamiento de datos en vectores:
    Nombre del modelo del producto (en formato textual).
    Precio unitario (debe ser mayor a cero).
    Cantidad solicitada, ídem anterior
-Validaciones especiales:
-Convertir el nombre del modelo automáticamente a mayúsculas.
-Si no se ingresan datos, el programa debe emitir un mensaje.
-Condición de corte: al ingresar "FIN" como nombre del modelo.
a) Mostrar todos los productos y sus datos asociados en formato de tabla, utilizando una función.
b) Identificar y mostrar el nombre del producto con el precio más bajo.
c) Identificar y mostrar todos los datos de la mayor cantidad solicitada
d) Calcular la cantidad promedio de todos los productos.
e) Eliminar los productos cuyo precio supere el promedio del punto anterior, junto con sus respectivos valores.
f) Insertar el modelo "PRUEBA" con un precio de 9999 y cantidad 7777, inmediatamente después de cada precio par.
g) Ordenar los productos de acuerdo a las cantidades, asegurando que acompamen el orden el resto de los datos."""

def cargar_datos(arre_mod, arre_pre, arre_cant):
    modelo = input("\nIngrese el nombre (FIN para terminar): ").upper()
    while modelo != "FIN":
        # Se corrigió validación para que precio sea > 0 (si es <= 0 pide de nuevo)
        precio = int(input("Ingrese el precio: "))
        while precio <= 0:
            precio = int(input("Error - Ingrese el precio (mayor a 0): "))

        cantidad = int(input("Ingrese la cantidad: "))
        while cantidad <= 0:
            cantidad = int(input("Error - Ingrese la cantidad (mayor a 0): "))

        arre_mod.append(modelo)
        arre_pre.append(precio)
        arre_cant.append(cantidad)
        modelo = input("\nIngrese el nombre (FIN para terminar): ").upper()

def mostrar_datos(arre_mod, arre_pre, arre_cant):
    titulo1 = "Modelo"
    titulo2 = "Precio"
    titulo3 = "Cantidad"
    print(f"{titulo1:<10}{titulo2:<8}{titulo3:>5}")
    print("-"*30)
    for i in range(len(arre_mod)):
        print(f"{arre_mod[i]:<10}{arre_pre[i]:<8}{arre_cant[i]:5}")
    print("-"*30)

def buscar_minimo(arre_pre):
    pos_min = 0
    minimo = arre_pre[pos_min]
    # Se implementó el ciclo for que faltaba
    for i in range(1, len(arre_pre)):
        if arre_pre[i] < minimo:
```

```
            minimo = arre_pre[i]
            pos_min = i
    return pos_min

def buscar_maximo(arre_cant):
    pos_max = 0
    maximo = arre_cant[pos_max]

    for i in range(1, len(arre_cant)):
        if arre_cant[i] > maximo:
            maximo = arre_cant[i] # Corregido: asignación (=), no comparación (<)
            pos_max = i
    return pos_max

def calcular_promedio(arre_datos):
    suma = 0
    for i in range(len(arre_datos)):
        suma += arre_datos[i]
    promedio = suma / len(arre_datos)
    return promedio

def eliminar_productos(arre_mod, arre_pre, arre_cant, promedio):
    # Se usa while para controlar el índice manualmente al borrar
    i = 0
    while i < len(arre_pre):
        # Corregido: Comparar el elemento [i] contra el promedio
        if arre_pre[i] > promedio:
            arre_mod.pop(i)
            arre_pre.pop(i)
            arre_cant.pop(i)
        else:
            i += 1

def insertar_productos(arre_mod, arre_pre, arre_cant):
    i = 0
    while i < len(arre_pre):
        # Corregido nombre de variable arre_pre
        if arre_pre[i] % 2 == 0:
            arre_mod.insert(i + 1, "PRUEBA")
            arre_pre.insert(i + 1, 9999) # Insertar número, no string
            arre_cant.insert(i + 1, 7777) # Insertar número, no string
            i += 2
        else:
            i += 1

def arre_aux(arreglo, i, j):
    auxiliar = arreglo[i]
    arreglo[i] = arreglo[j]
    arreglo[j] = auxiliar

def ordenar_datos(arre_mod, arre_pre, arre_cant):
    for i in range(len(arre_cant) - 1):
        for j in range(i + 1, len(arre_cant)):
            if arre_cant[i] > arre_cant[j]:
                arre_aux(arre_cant, i, j)
                arre_aux(arre_mod, i, j)
                arre_aux(arre_pre, i, j)
```

```python
# --- PROGRAMA PRINCIPAL ---
print("-"*30)
modelos = []
precios = []
cantidades = []

cargar_datos(modelos, precios, cantidades)

if len(modelos) > 0:
    print("\na) Mostrar todos los productos: \n")
    mostrar_datos(modelos, precios, cantidades)

    print("\nb) Mostrar el precio más bajo: \n")
    pos_min = buscar_minimo(precios)
    print(f"El modelo {modelos[pos_min]} con un precio de {precios[pos_min]}.")

    print("\nc) Mostrar la cantidad mayor: \n")
    # buscar_maximo devuelve la POSICIÓN
    pos_max = buscar_maximo(cantidades)
    print(f"{cantidades[pos_max]} unidades del modelo {modelos[pos_max]} con valor de
{precios[pos_max]}.")

    print("\nd) Calcular el promedio de los productos:\n")
    # El enunciado pide "cantidad promedio" (punto d) Pero el punto (e) pide eliminar por
PRECIO mayor al promedio ANTERIOR.
    # Hay una contradicción en el enunciado. Calcularemos el promedio de PRECIOS para que
el punto (e) tenga sentido lógico.
    promedio_precios = calcular_promedio(precios)
    print(f"Promedio de precios: {promedio_precios:.2f}")

    print("\ne) Eliminar superiores al promedio: \n")
    eliminar_productos(modelos, precios, cantidades, promedio_precios)
    mostrar_datos(modelos, precios, cantidades)

    print("\nf) Insertar después de cada precio par: \n")
    print("modelo: PRUEBA, precio: 9999 y cantidad: 7777\n")
    insertar_productos(modelos, precios, cantidades)
    mostrar_datos(modelos, precios, cantidades)

    print("\ng) Ordenar por cantidades: \n")
    ordenar_datos(modelos, precios, cantidades)
    mostrar_datos(modelos, precios, cantidades)
else:
    print("\nNo se ingresaron datos... ")
print("\nFin del programa.")
print("-"*30)
```