

""Carla, queda seleccionada como gerente de una empresa de informática, se encuentra con la solicitud de su equipo de desarrollar un sistema de registro de stock. Para abordar esta necesidad, le encomienda la tarea de crear un sistema utilizando los siguientes datos:

Se debe almacenar la siguiente información:

- Nombre del producto
- Importe del producto
- Cantidad de stock

Es importante destacar que la cantidad de stock no puede ser negativa. Además, los nombres de los productos deben ser convertidos a mayúscula.

La carga de datos se efectúa en tres vectores: uno para los nombres de los productos, otro para los importes, y otro para el stock.

La carga finaliza cuando se ingresa "CARLA" como nombre del producto. Se deben aplicar validaciones lógicas. Presentar todos los datos en formato de tabla después de cada punto donde se modifiquen los vectores. Tareas a realizar:

1. Calcular la cantidad de stock promedio
2. Mostrar el nombre del producto con la mayor cantidad de stock de manera programática (sin ordenar).
3. Eliminar los datos (producto, importe y cant. stock) cuya cantidad de stock sea par y mayor de 36.
4. Reemplazar con los valores : producto = " reemplazo", importe = " 0" y stock ="-5) cuando el stock sea entre 15 y 35
5. Insertar en la posición siguiente de cada importe múltiplo de 5 los siguientes valores: producto "INSERTADO", VALOR 0, y STOCK -1.
6. Ordenar utilizando el método burbuja según las cantidades de stock, de mayor a menor.""

```
# --- FUNCIONES DE UTILIDAD ---
```

```
# --- 1. Carga de Datos y Validaciones ---def cargar_pedidos(arr_mod, arr_cant):
```

```
def cargar_datos(arr_nombres, arr_importes, arr_stock):
    """Carga los datos de productos con validaciones."""

    print("\n--- INGRESO DE DATOS DE STOCK ---")
    nombre = input("Ingrese Nombre del producto (o 'CARLA' para finalizar): ").upper()
    while nombre != "CARLA":
        while nombre == "":
            nombre = input("Error - Ingrese Nombre del producto (o 'CARLA' para
finalizar): ").upper()
            importe = float(input(f"Importe del producto '{nombre}': $"))
            while importe < 0:
                importe = float(input(f"Error - Importe del producto '{nombre}': $"))
            cantidad = int(input(f"Cantidad de stock para '{nombre}': "))
            while cantidad < 0:
                cantidad = int(input(f"Error - Cantidad de stock para '{nombre}': "))

            arr_nombres.append(nombre)
            arr_importes.append(importe)
            arr_stock.append(cantidad)
            nombre = input("Ingrese Nombre del producto (o 'CARLA' para finalizar):
").upper()
```

```
def mostrar_tabla(arr_nombres, arr_importes, arr_stock):
    """Muestra los datos almacenados en formato de tabla."""
    titulo1 = "Producto"
    titulo2 = "Importe"
    titulo3 = "Stock"
    print(f"\n{titulo1:<20}{titulo2:<10}{titulo3}")
    print("-----")
```

```

for i in range(len(arr_nombres)):
    print(f"{arr_nombres[i]:<20}{arr_importes[i]:<10}{arr_stock[i]}")
print("-----")

# --- TAREAS SOLICITADAS ---

# 1. Calcular la cantidad de stock promedio
def calcular_stock_promedio(arr_stock):
    """Calcula el promedio de la cantidad de stock."""
    total_stock = 0
    for i in range(len(arr_stock)):
        if arr_stock[i] and arr_stock[i] > 0:
            total_stock += arr_stock[i]
    return total_stock / len(arr_stock)

# 2. Mostrar el nombre del producto con la mayor cantidad de stock
def buscar_maximo_stock(arr):
    """Busca el índice del producto con mayor stock."""
    pos_max = 0
    for i in range(1, len(arr)):
        if arr[i] > arr[pos_max]:
            pos_max = i
    return pos_max

# 3. Eliminar los datos cuya cantidad de stock sea par y mayor de 36
def eliminar_stock_par_mayor_36(arr_nombres, arr_importes, arr_stock):
    """Elimina elementos cuyo stock sea par y mayor a 36."""
    i = 0

    while i < len(arr_stock):
        # Condición: stock PAR y stock > 36
        if arr_stock[i] % 2 == 0 and arr_stock[i] > 36:
            arr_nombres.pop(i)
            arr_importes.pop(i)
            arr_stock.pop(i)
            # NO incrementamos 'i' porque el siguiente elemento ocupa esta posición
        else:
            i += 1

# 4. Reemplazar cuando el stock sea entre 15 y 35 (ambos inclusive)
def reemplazar_stock_entre_15_y_35(arr_nombres, arr_importes, arr_stock):
    """Reemplaza valores si el stock está entre 15 y 35 (inclusive)."""
    for i in range(len(arr_stock)):
        # Condición: Stock >= 15 y Stock <= 35
        if 15 <= arr_stock[i] <= 35:
            arr_nombres[i] = "REEMPLAZO"
            arr_importes[i] = "0"
            arr_stock[i] = -5

# 5. Insertar después de cada importe múltiplo de 5
def insertar_por_importe_mult5(arr_importes, arr_nombres, arr_stock):
    """Inserta nuevos valores en la posición siguiente de cada importe múltiplo de 5."""
    i = 0
    valor_entero = int(float(arr_importes[i]))
    while i < len(arr_importes):
        if valor_entero % 5 == 0: # Si es múltiplo de 5, insertamos
            arr_nombres.insert(i + 1, "INSERTADO")
            arr_importes.insert(i + 1, "0")
            arr_stock.insert(i + 1, -1)
            i += 2 # Avanzamos 'i' al nuevo elemento insertado
        else:

```

```

        i += 1

# Función de intercambio (intercambiar) para ordenar varios vectores
def intercambiar(arr_stock, i, j):
    aux = arr_stock[i]
    arr_stock[i] = arr_stock[j]
    arr_stock[j] = aux

# 6. Ordenar por stock de mayor a menor (Método Burbuja)
def ordenar_stock_burbuja(arr_stock, arr_importes, arr_nombres):
    for i in range(len(arr_stock) - 1):
        for j in range(i + 1, len(arr_nombres)):
            if arr_stock[i] < arr_importes[j]:
                intercambiar(arr_stock, i, j)
                intercambiar(arr_nombres, i, j)
                intercambiar(arr_importes, i, j)

# ----- PROGRAMA PRINCIPAL -----
print("\n-----")
print("      SISTEMA DE GESTIÓN DE STOCK (CARLA)")
print("-----")

nombres = []
importes = []
stock = []

# Carga de Datos
cargar_datos(nombres, importes, stock)
print("\n-----")

if len(nombres) > 0:
    mostrar_tabla(nombres, importes, stock)
    print("\n-----")
    # 1. Stock Promedio
    promedio = calcular_stock_promedio(stock)
    print(f"1. Stock Promedio: {promedio:.2f}")
    print("\n-----")
    # 2. Producto con Mayor Stock
    index_max = buscar_maximo_stock(stock)
    print(f"2. Producto con Mayor Stock: {nombres[index_max]} (Stock: {stock[index_max]})")
    print("\n-----")
    # 3. Eliminar stock par y > 36
    eliminar_stock_par_mayor_36(nombres, importes, stock)
    print(f"3. Eliminar productos (Stock par > 36).")
    mostrar_tabla(nombres, importes, stock)
    print("\n-----")
    # 4. Reemplazar stock entre 15 y 35
    reemplazar_stock_entre_15_y_35(nombres, importes, stock)
    print(f"4. Reemplazar productos (Stock entre 15 y 35).")
    mostrar_tabla(nombres, importes, stock)
    print("\n-----")
    # 5. Insertar después de importe múltiplo de 5
    insertar_por_importe_mult5(importes, nombres, stock)
    print(f"5. Insertar productos (Importe Múltiplo de 5).")
    mostrar_tabla(nombres, importes, stock)
    print("\n-----")
    # 6. Ordenar por stock de mayor a menor (Burbuja)
    ordenar_stock_burbuja(stock, importes, nombres)
    print("6. Datos Ordenados por Stock (Mayor a Menor).")
    mostrar_tabla(nombres, importes, stock)

```

```
else:
    print("No se ingresaron productos.")
print("\n-----")
print("Fin del programa...")
print("-----\n")
```