

#Segundo Parcial Modelo 2 del jueves

"""1. La realización de un sistema con los datos que se brindan a continuación. Se leen los siguientes datos:

-Nombre del clientes (Finaliza cuando se coloca "FIN")

-Nombre del contador asignado

-Importe obtenido como ganancia anual (debe ser menor o igual a 0), en ese caso se debe solicitar el valor nuevamente.

La carga se realiza en 3 vectores, (clientes, nombre del contador y el importe). Realizar validaciones con criterios lógicos.

1. Mostrar todos los datos.

2. Calcular el importe promedio de todos los clientes.

3. Mostrar los 3 arreglos ordenados por el importe obtenido, de menor a mayor.

4. Mostrar el nombre del cliente de mayor importe obtenido como ganancia actual.

5. Insertar despues de cada importe impar superior al promedio de los siguientes valores:

cliente: "CLIENTE", contador: "CONTADOR", importe: -9999

6. Eliminar los datos (clientes, contadores e importes) cuya ganancia sea par y menor al promedio"""

#FUNCIONES

```
def validar_contador():
```

```
    contador_nom = input("Ingrese el nombre del contador asignado: ").upper()
```

```
    while contador_nom == "":
```

```
        contador_nom = input("Error. Ingrese el nombre del contador asignado: ").upper()
```

```
    return contador_nom
```

```
def validar_importe():
```

```
    importe_ent = int(input("Ingrese el importe (debe ser mayor a 0): "))
```

```
    while importe_ent <= 0:
```

```
        importe_ent = int(input("Error. Ingrese el importe (debe ser mayor a 0): "))
```

```
    return importe_ent
```

```
def cargar_datos(arre_cliente,arre_contadore,arre_importe):
```

```
    cliente = input("Ingrese el nombre (o 'FIN' para terminar): ").upper()
```

```
    while cliente != "" and cliente != "FIN":
```

```
        contador = validar_contador()
```

```
        importe = validar_importe()
```

```
        arre_cliente.append(cliente)
```

```
        arre_contadore.append(contador)
```

```
        arre_importe.append(importe)
```

```
        cliente = input("\nIngrese el nombre (o 'FIN' para terminar): ").upper()
```

```
    print("-----")
```

```
def mostrar_datos(arre_cliente,arre_contador,arre_importe):
```

```
    titulo1 = "Cliente"
```

```
    titulo2 = "Contador"
```

```
    titulo3 = "Importe"
```

```
    print(f"{titulo1:<15}{titulo2:<15}{titulo3:>10}")
```

```
    print("-----")
```

```
    for i in range(len(arre_cliente)):
```

```
        print(f"{arre_cliente[i]:<15}{arre_contador[i]:<10}{arre_importe[i]:>10}")
```

```
    print("-----\n")
```

```
def calcular_promedio(arre_importe):
```

```
    acum = 0
```

```
    cont = 0
```

```

for i in range(len(arre_importe)):
    acum += arre_importe[i]
    cont += 1
promedio = acum/cont
return promedio

def intercambiar(arre_importe, i, j):
    auxiliar=arre_importe[i]
    arre_importe[i] = arre_importe[j]
    arre_importe[j] = auxiliar

def ordenar_datos(arre_cliente, arre_contador, arre_importe):
    for i in range(len(arre_importe)-1):
        for j in range(i + 1,len(arre_importe)):
            if arre_importe[i] > arre_importe[j]:
                intercambiar(arre_importe, i, j)
                intercambiar(arre_cliente, i, j)
                intercambiar(arre_contador, i, j)

def mayor_importe(arre_importe):
    pos_max = 0
    maximo = arre_importe[pos_max]

    for i in range(1, len(arre_importe)):
        if arre_importe[i] > maximo:
            maximo = arre_importe[i]
            pos_max = i
    return pos_max

def insertar_datos(arre_cliente,arre_contador,arre_importe):
    valor_promedio=calcular_promedio(arre_importe)
    i = 0
    while i < len(arre_importe):
        es_impar = arre_importe[i] % 2 != 0
        es_superior = arre_importe[i] > valor_promedio
        if es_impar and es_superior:
            # Insertar después de la posición i
            arre_cliente.insert(i + 1, "CLIENTE")
            arre_contador.insert(i + 1, "CONTADOR")
            arre_importe.insert(i + 1, -9999)
            i += 2 # Saltar el insertado
        else:
            i += 1

def eliminar_datos(arre_cliente,arre_contador,arre_importe,promedio):
    i = 0
    while i < len(arre_importe):
        es_par = arre_importe[i] % 2 == 0
        es_menor = promedio > arre_importe[i]
        if es_par and es_menor:
            arre_cliente.pop(i)
            arre_contador.pop(i)
            arre_importe.pop(i)
        else:
            i += 1

#PPAL
clientes = []

```

```

contadores = []
importes = []
print("\n-----")
cargar_datos(clientes, contadores, importes)

if len(clientes) > 0:

    print("1. Mostrar todos los datos:\n")
    mostrar_datos(clientes, contadores, importes)
    print("2. Promedio del importe de todos los clientes:\n")
    valor_promedio=calcular_promedio(importes)
    print(f"Importe promedio: {valor_promedio:.2f}")
    print("-----\n")
    print("3. Datos ordenados por importe ascendente:\n")
    ordenar_datos(clientes, contadores, importes)
    mostrar_datos(clientes, contadores, importes)
    print("4. Cliente con mayor importe:\n")
    index_max= mayor_importe(importes)
    print(f"Cliente: {clientes[index_max]} con un importe de {importes[index_max]:.2f}")
    print("-----\n")
    print("5. Datos después de insertar (impar>promedio):\n")
    print("cliente:CLIENTE, contador=CONTADOR, importe: -9999\n")
    insertar_datos(clientes,contadores,importes)
    mostrar_datos(clientes, contadores, importes)
    print("6. Datos después de eliminar (par<promedio):\n")
    eliminar_datos(clientes,contadores,importes,valor_promedio)
    mostrar_datos(clientes, contadores, importes)
else:
    print("No se ingresaron datos...")
print("Fin del programa.")
print("-----\n")

```