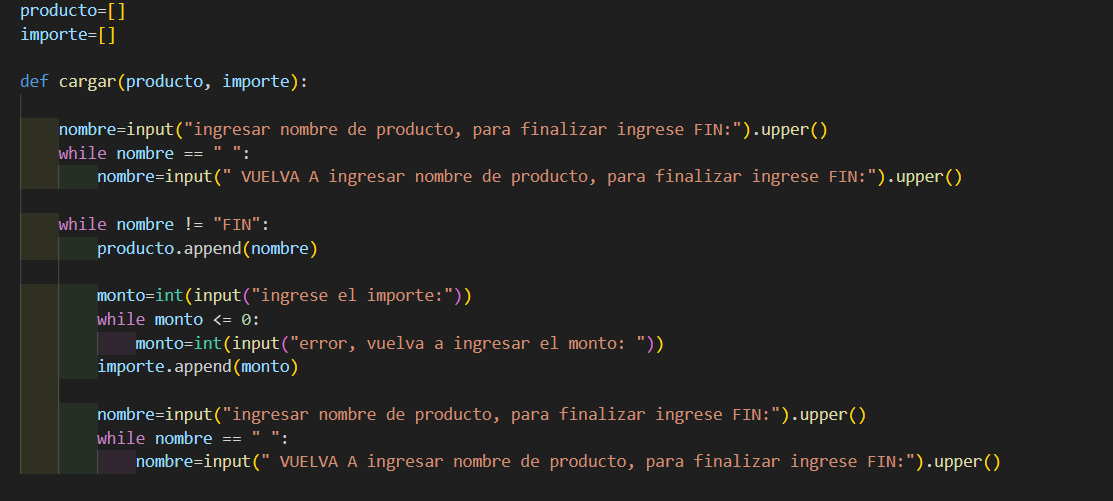
**Aperturas para codigo**

****

• Diseñar un algoritmo que permita ingresar nombre de productos e importes (2 vectores). • Esta carga de productos finaliza con el nombre de producto "FIN". • El importe a cargar no puede ser menor o igual 0.

Para realizar una estadística la empresa "El estadista", le pide desarrollar un sistema que procese los datos solicitados en una encuesta.

Para ello se carga la edad, cantidad de hijos y la provincia donde vive (se codifican de 1 a 10).

La carga finaliza cuando se ingresa 0 en la edad o se hayan procesados mas de 1000 encuestas

**A computer screen shot of a program code

Description automatically generated**

**Ejemplo de examen**

# Se leen y se cargan en vectores los siguientes datos:

# Modelo de producto, en letras

# Precio unitario, no puede ser negativos ni 0

# Cabe destacar además, que el nombre del modelo puede ingresarse en mayúscula o minúscula.

# En el caso de que se ingrese en minúscula, se debe convertir a mayúscula.

# Si la carga es nula (no se ingresaron datos), mostrar una leyenda y finalizar la ejecución.

# La carga de productos finaliza cuando se coloca “FIN” en el nombre del producto.

# 1) Mostrar en forma de lista de productos y sus precios en una sola función.

# 2) Mostrar el nombre del producto de mayor precio

# 3) Calcular el promedio de los precios

# 4) Eliminar los valores que superen al promedio y sus productos asociados.

# 5) Ordenar ambos vectores, elegir el criterio por precio o producto.

# Recuerden que luego de insertar, eliminar u ordenar se debe mostrar cómo quedan los vectores

def cargarDatos(modelo, precio):

    modeloProducto = input("INGRESE NOMBRE DEL PRODUCTO (\"FIN\" para terminar): ").upper()

    while modeloProducto != "FIN":

        modelo.append(modeloProducto)

        precioProducto = int(input("INGRESE EL PRECIO DEL PRODUCTO: "))

        while precioProducto <= 0:

            precioProducto = int(input("ERROR - INGRESE EL PRECIO DEL PRODUCTO: "))

        precio.append(precioProducto)

        modeloProducto = input("INGRESE NOMBRE DEL PRODUCTO (\"FIN\" para terminar): ").upper()

# 1) Mostrar en forma de lista de productos y sus precios en una sola función.

def mostrar(modelo, precio, mensaje, mensaje2):

    print(mensaje2)

    print(mensaje)

    for i in range(len(precio)):

        print(f"{modelo[i]} - {precio[i]}")

# 2) Mostrar el nombre del producto de mayor precio

def mayorPrecio(precio):

    aux = precio[0]

    pos = 0

    for i in range(len(precio)):

        if precio[i] > aux:

            aux = precio[i]

            pos = i

    return pos

# 3) Calcular el promedio de los precios

def promedio(precio):

    sum = 0

    cont = 0

    for i in range(len(precio)):

        sum += precio[i]

        cont += 1

    prom = sum/cont

    return prom

# 4) Eliminar los valores que superen al promedio y sus productos asociados.

def eliminar(modelo, precio, promedio):

    i = 0

    while i < len(precio):

        if precio[i] > promedio:

            precio.pop(i)

            modelo.pop(i)

        else:

            i+=1

# 5) Ordenar ambos vectores, elegir el criterio por precio o producto.

def ordenar\_mayor(precio, modelo):

    for i in range(0,len(precio)-1):

        for j in range(i +1 , len(precio)):

            if precio[i] > precio[j]:

                    swap(precio, i, j)

                    swap(modelo, i, j)

def swap(vec, i, j):

    aux = vec[i]

    vec[i] = vec[j]

    vec[j] = aux

# PROGRAMA PRINCIPAL

modelo = []

precio = []

cargarDatos(modelo, precio)

if len(modelo) > 0:

    mostrar(modelo, precio, "ARREGLOS EN FORMA DE LISTA\n","\n##### PRIMER EJERCICIO #####")

    print("\n##### SEGUNDO EJERCICIO #####")

    print("NOMBRE DEL PRODUCTO DE MAYOR PRECIO\n")

    print(f"PRODUCTO: {modelo[mayorPrecio(precio)]}, PRECIO: {precio[mayorPrecio(precio)]}\n")

    print("##### TERCER EJERCICIO #####")

    print("CALCULAR PROMEDIO DE LOS PRECIOS\n")

    print(f"EL PROMEDIO DE LOS PRECIOS ES: {promedio(precio)}\n")

    eliminar(modelo,precio,promedio(precio))

    mostrar(modelo, precio, "ARREGLOS CON VALORES SUPERIORES AL PROMEDIO ELIMINADOS\n","##### CUARTO EJERCICIO #####")

    ordenar\_mayor(precio, modelo)

    mostrar(modelo, precio, "ARREGLOS ORDENADO DE MAYOR A MENOR\n","\n##### QUINTO EJERCICIO #####")

else:

    print("\n####### NO SE INGRESARON DATOS. FIN DEL PROGRAMA ######\n")