



Tecnológico de Monterrey

Maestría en Inteligencia Artificial Aplicada - Programa de Posgrados en Línea

Actividad 5.2 Ejercicios de programación 2

A01795690. Michelle Alejandro Durán Sánchez

Escuela de Ingeniería y Ciencias, Tecnológico de Monterrey

TC4017.10: Pruebas de software y aseguramiento de la calidad

Dr. Gerardo Padilla Zárate

Mtra. María Mylen Treviño E.

Ciudad de México, 09 de febrero de 2025

Ejercicios de programación

Ejercicio 1: Compute Sales

Requerimiento:

Requisito 1. El programa deberá ejecutarse desde una línea de comandos. El programa deberá recibir dos archivos como parámetros. El primer archivo contendrá información en formato JSON sobre un catálogo de precios de productos. El segundo archivo contendrá un registro de todas las ventas en una empresa.

Requisito 2. El programa deberá calcular el costo total de todas las ventas incluidas en el segundo archivo JSON. Los resultados deberán imprimirse en pantalla y guardarse en un archivo llamado *SalesResults.txt*. El costo total debe incluir todos los elementos de la venta considerando el costo de cada artículo en el primer archivo. La salida debe ser legible para los humanos, por lo que debe ser fácil de leer para el usuario.

Requisito 3. El programa deberá incluir un mecanismo para manejar datos inválidos en el archivo. Los errores deben mostrarse en la consola y la ejecución debe continuar.

Requisito 4. El nombre del programa deberá ser *computeSales.py*.

Requisito 5. El formato mínimo para invocar el programa deberá ser el siguiente:

- `python computeSales.py priceCatalogue.json salesRecord.json`

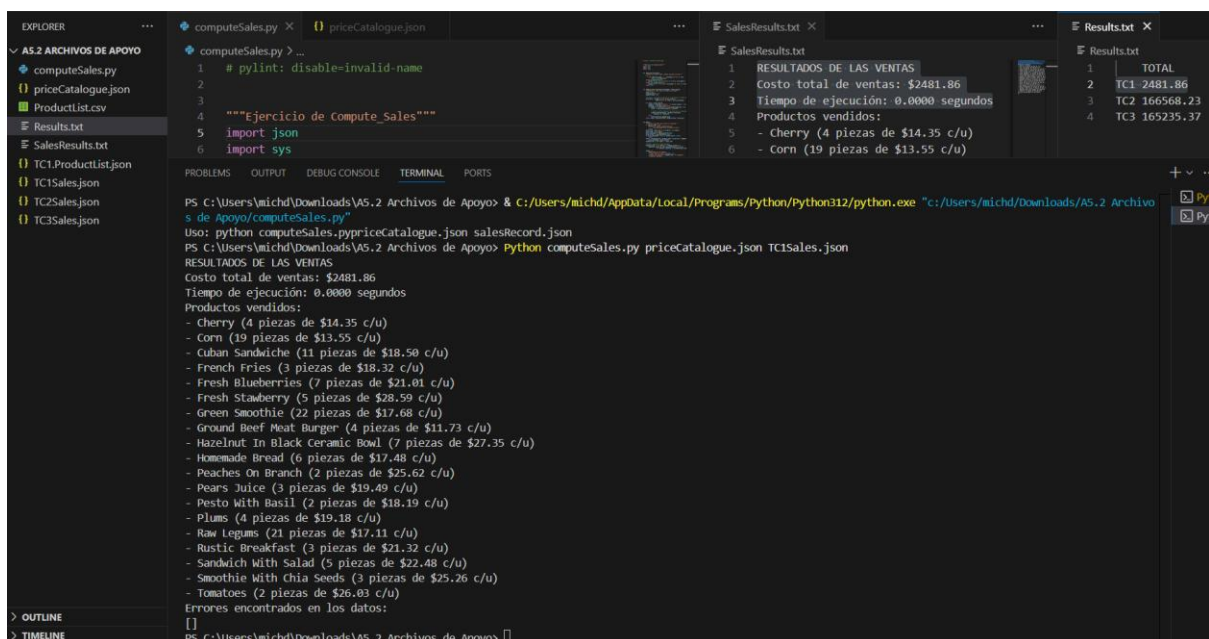
Requisito 6. El programa deberá manejar archivos que contengan desde cientos hasta miles de elementos.

Requisito 7. El programa deberá incluir al final de la ejecución el tiempo transcurrido para la ejecución y cálculo de los datos. Este número deberá incluirse tanto en el archivo de resultados como en la pantalla.

Requisito 8. El programa deberá cumplir con el estándar PEP8

Resultados:

- Suma total con Lista de productos de **TC1Sales.json** (\$2,481.86):



```
PS C:\Users\michd\Downloads\AS.2 Archivos de Apoyo> & C:\Users\michd\AppData\Local\Programs\Python\Python312\python.exe "c:/Users/michd/Downloads/AS.2 Archivos de Apoyo/computeSales.py" priceCatalogue.json salesRecord.json
Usó: python computeSales.py priceCatalogue.json salesRecord.json
PS C:\Users\michd\Downloads\AS.2 Archivos de Apoyo> python computeSales.py priceCatalogue.json TC1Sales.json
RESULTADOS DE LAS VENTAS
Costo total de ventas: $2481.86
Tiempo de ejecución: 0.0000 segundos
Productos vendidos:
- Cherry (4 piezas de $14.35 c/u)
- Corn (19 piezas de $13.55 c/u)
- Cuban Sandwich (11 piezas de $18.50 c/u)
- French Fries (3 piezas de $18.32 c/u)
- Fresh Blueberries (7 piezas de $21.01 c/u)
- Fresh Strawberry (5 piezas de $28.59 c/u)
- Green Smoothie (22 piezas de $17.68 c/u)
- Ground Beef Meat Burger (4 piezas de $11.73 c/u)
- Hazelnut In Black Ceramic Bowl (7 piezas de $27.35 c/u)
- Homemade Bread (6 piezas de $17.48 c/u)
- Peaches On Branch (2 piezas de $25.62 c/u)
- Pears Juice (3 piezas de $19.49 c/u)
- Pesto with Basil (2 piezas de $18.19 c/u)
- Plums (4 piezas de $19.18 c/u)
- Raw Legums (21 piezas de $17.11 c/u)
- Rustic Breakfast (3 piezas de $21.32 c/u)
- Sandwich with Salad (5 piezas de $22.48 c/u)
- Smoothie with Chia Seeds (3 piezas de $25.26 c/u)
- Tomatoes (2 piezas de $26.03 c/u)
Errores encontrados en los datos:
[]
PS C:\Users\michd\Downloads\AS.2 Archivos de Apoyo>
```

| | TOTAL |
|---|---------------|
| 1 | TC1 2481.86 |
| 2 | TC2 166568.23 |
| 3 | TC3 165235.37 |

Imagen 1. Resultados de TC1Sales.json.

- Suma total con Lista de productos de **TC2Sales.json** (\$166,568.23):

```

1 # pylint: disable=invalid-name
2
3
4 """Ejercicio de Compute_Sales"""
5 import json
6 import sys
7 import time
8
9
PS C:\Users\michd\Downloads\AS.2 Archivos de Apoyo> & C:\Users\michd\AppData\Local\Programs\Python\Python312\python.exe "c:\Users\michd\Downloads\AS.2 Archivos de Apoyo\computeSales.py"
PS C:\Users\michd\Downloads\AS.2 Archivos de Apoyo> python computeSales.py priceCatalogue.json salesRecord.json
PS C:\Users\michd\Downloads\AS.2 Archivos de Apoyo> Python computeSales.py priceCatalogue.json TC2Sales.json
RESULTADOS DE LAS VENTAS
Costo total de ventas: $166568.23
Tiempo de ejecución: 0.0000 segundos
Productos vendidos:
- Corn (1664 piezas de $13.55 c/u)
- Cuban Sandwich (2 piezas de $18.50 c/u)
- French Fries (57 piezas de $18.32 c/u)
- Fresh Blueberries (3790 piezas de $21.01 c/u)
- Fresh Strawberry (221 piezas de $28.59 c/u)
- Green Smoothie (580 piezas de $17.68 c/u)
- Ground Beef Meat Burger (146 piezas de $11.73 c/u)
- Hazelnut In Black Ceramic Bowl (7 piezas de $27.35 c/u)
- Homemade Bread (466 piezas de $17.48 c/u)
- Plums (1231 piezas de $19.18 c/u)
- Raw Legums (13 piezas de $17.11 c/u)
- Rustic Breakfast (200 piezas de $21.32 c/u)
- Sandwich With Salad (23 piezas de $22.48 c/u)
- Smoothie With Chia Seeds (61 piezas de $25.26 c/u)
- Sweet Fresh Strawberry (221 piezas de $29.45 c/u)
- Tomatoes (1 piezas de $26.03 c/u)
Errores encontrados en los datos:
[]
  
```

Imagen 2. Resultados de TC2Sales.json.

- Suma total con Lista de productos de **TC3Sales.json** (\$166,568.23) y errores al no encontrar ciertos productos:

```

1 # pylint: disable=invalid-name
2
3
4 """Ejercicio de Compute_Sales"""
5 import json
6 import sys
7 import time
8
9
PS C:\Users\michd\Downloads\AS.2 Archivos de Apoyo> & C:\Users\michd\AppData\Local\Programs\Python\Python312\python.exe "c:\Users\michd\Downloads\AS.2 Archivos de Apoyo\computeSales.py"
PS C:\Users\michd\Downloads\AS.2 Archivos de Apoyo> python computeSales.py priceCatalogue.json salesRecord.json
PS C:\Users\michd\Downloads\AS.2 Archivos de Apoyo> Python computeSales.py priceCatalogue.json TC3Sales.json
RESULTADOS DE LAS VENTAS
Costo total de ventas: $165235.37
Tiempo de ejecución: 0.0000 segundos
Productos vendidos:
- Corn (1664 piezas de $13.55 c/u)
- Cuban Sandwich (2 piezas de $18.50 c/u)
- French Fries (37 piezas de $18.32 c/u)
- French Fries (37 piezas de $18.32 c/u)
- Fresh Blueberries (3744 piezas de $21.01 c/u)
- Fresh Strawberry (221 piezas de $28.59 c/u)
- Green Smoothie (580 piezas de $17.68 c/u)
- Ground Beef Meat Burger (146 piezas de $11.73 c/u)
- Hazelnut In Black Ceramic Bowl (7 piezas de $27.35 c/u)
- Homemade Bread (466 piezas de $17.48 c/u)
- Plums (1231 piezas de $19.18 c/u)
- Raw Legums (13 piezas de $17.11 c/u)
- Rustic Breakfast (200 piezas de $21.32 c/u)
- Sandwich With Salad (23 piezas de $22.48 c/u)
- Smoothie With Chia Seeds (61 piezas de $25.26 c/u)
- Sweet Fresh Strawberry (221 piezas de $29.45 c/u)
- Tomatoes (1 piezas de $26.03 c/u)
Errores encontrados en los datos:
[
  {
    "SALE_ID": 6,
    "SALE_Date": "01/12/23",
    "Product": "Elotes",
    "Quantity": 100
  },
  {
    "SALE_ID": 9,
    "SALE_Date": "02/12/23",
    "Product": "Frijoles",
    "Quantity": 100
  }
]
  
```

Imagen 3. Resultados de TC3Sales.json.

- No existen “**Elotes**” (imagen 4) ni “**Frijoles**” (imagen 5) en el Catálogo de precios:

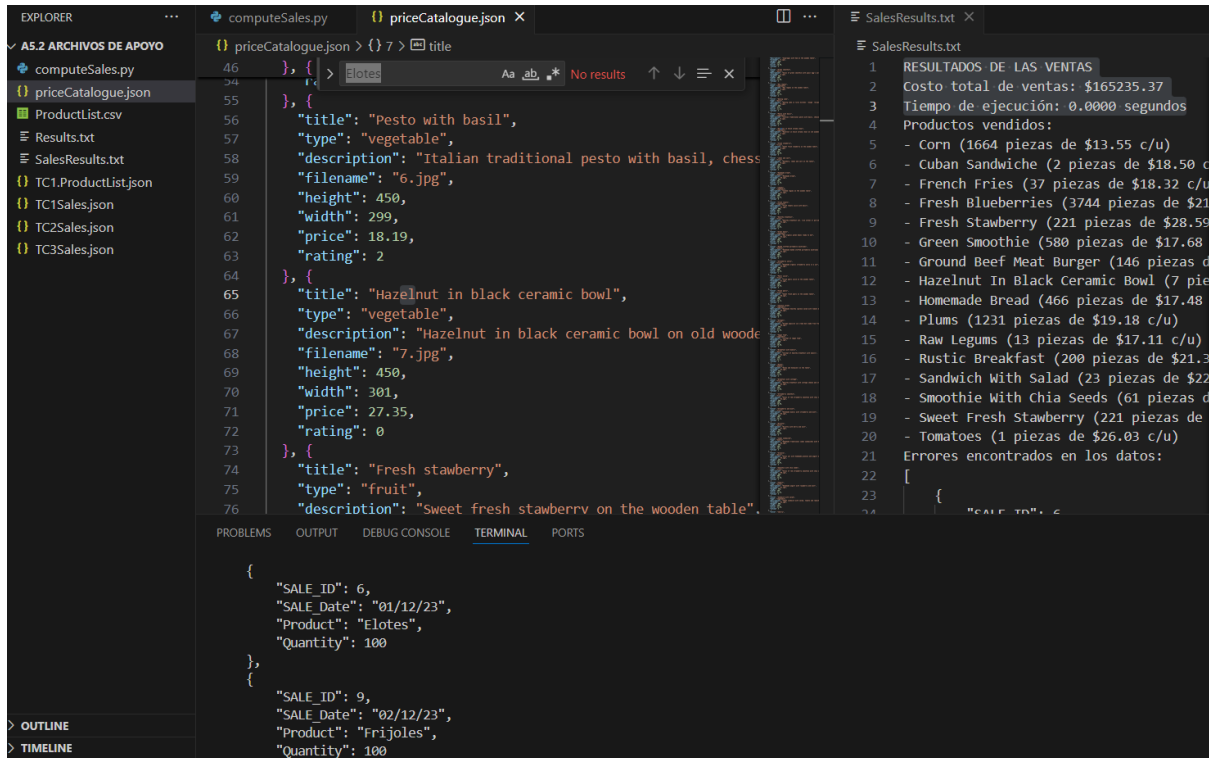


Imagen 4. No existen “Elotes”

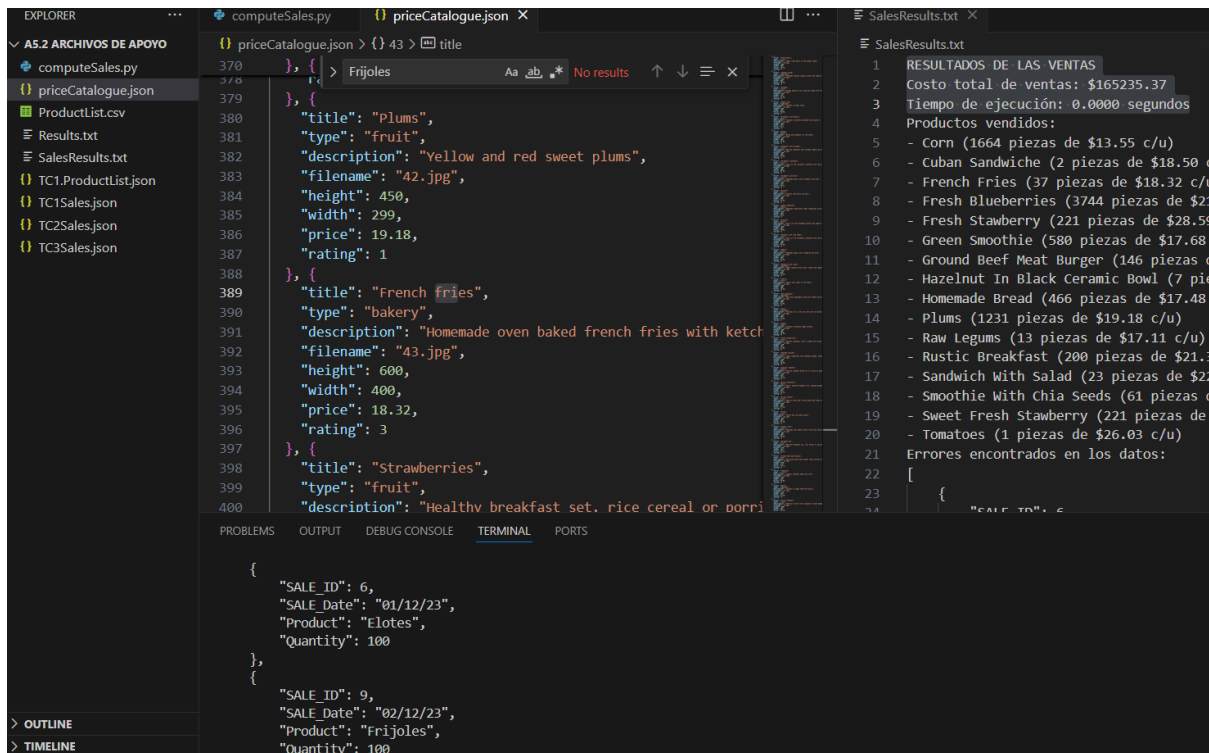


Imagen 5. No existen “Frijoles”

- Muestra de validación con **Pylint** sin errores en la imagen 6. Si los hubiera, se vería como en la imagen 7.

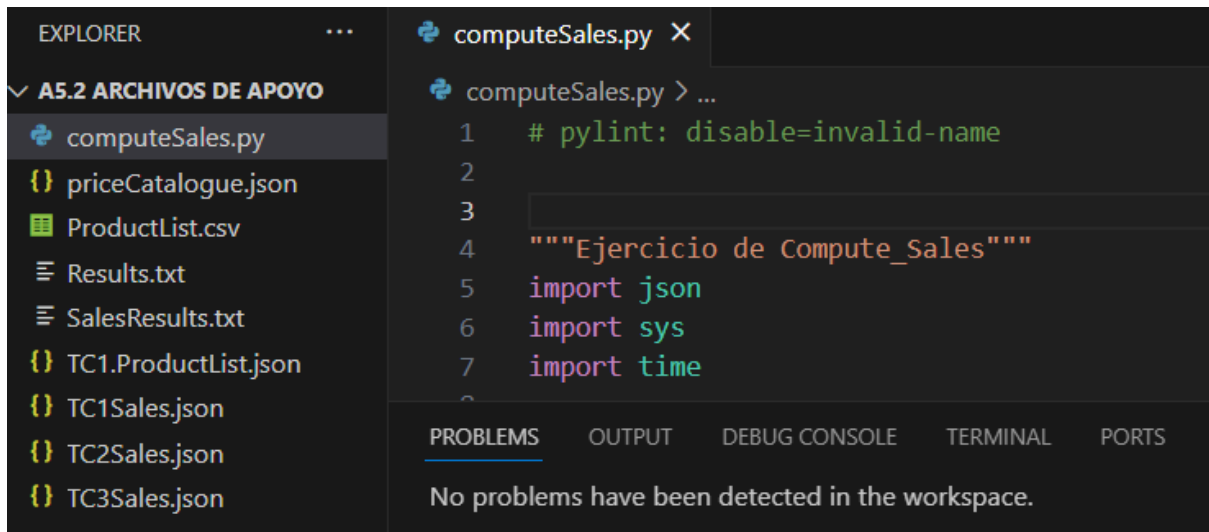


Imagen 6. Sin errores.

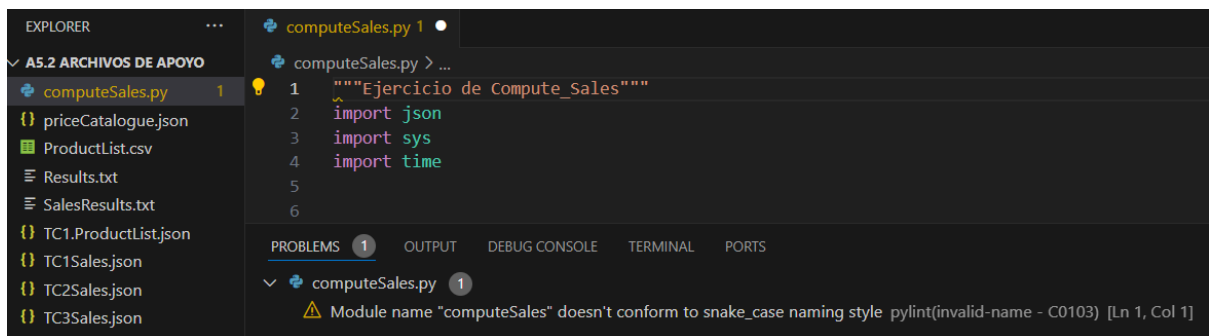


Imagen 7. Con warning de que el nombrado no cumple con la nomenclatura.

- Muestra de validación con **Flake8** sin errores en la imagen 8. Si los hubiera, se vería como en la imagen 9.

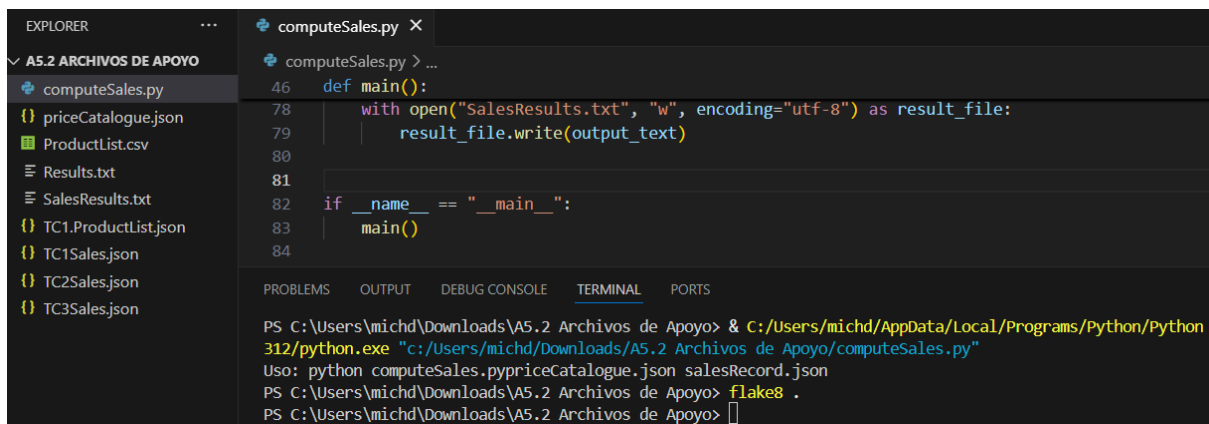


Imagen 8. Sin errores.

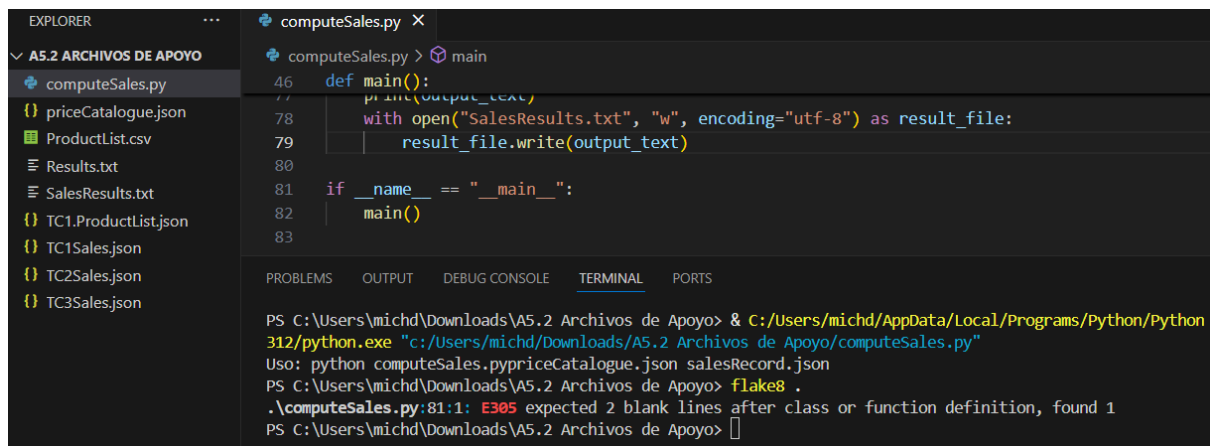


Imagen 9. Espera tener 2 espacios en blanco después de la función, pero sólo encontró 1.

Liga de GitHub para ver ejercicios:

- https://github.com/MichelleDuranTEC/A01795690_A5.2.git

Referencias Bibliográficas

Píldoras de programación. (2023, 26 de marzo). *Flake8 en Python, rápido y sencillo 2023*. [Video]. YouTube. <https://www.youtube.com/watch?v=G3tSdlBnSgA>