EX:No.7 DATE:29/03/25	**Implement program for decomposing time series data into trend and seasonality**

AIM:

To implement a program that decomposes a given time series dataset into its trend, seasonal, and residual components using the seasonal decomposition method, and visualize each component to better understand the underlying patterns in the electric production data.

ALGORITHM:

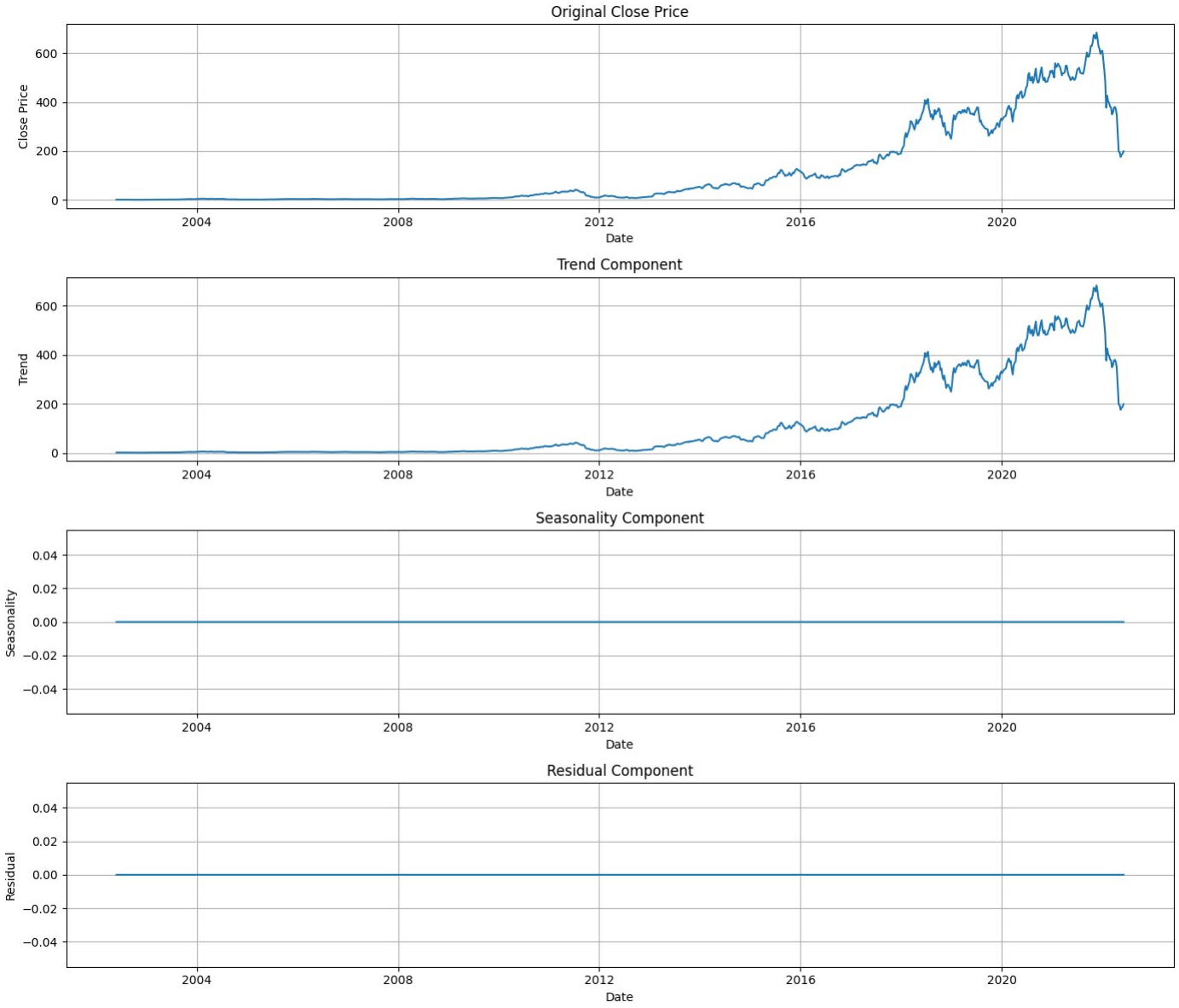
1. Import the required libraries: pandas for data manipulation, matplotlib.pyplot for plotting, and seasonal_decompose from statsmodels for time series decomposition.
2. Load the electric production dataset and convert the date column to datetime format.
3. Set the date column as the index to enable time series operations.
4. Apply seasonal decomposition using an additive model with a defined period (e.g., 12 months for yearly seasonality in monthly data).
5. Extract the decomposed components: trend, seasonality, and residual from the result.
6. Plot the original series along with the trend, seasonal, and residual components in a single figure using subplots for visual analysis.

CODE:

```
 import pandas as pd
from statsmodels.tsa.seasonal import seasonal_decompose
import matplotlib.pyplot as plt

# Load the data
df = pd.read_csv('NFLX (1).csv', parse_dates=['Date'], index_col='Date')

# Resample to weekly frequency
df_weekly = df.resample('W').agg(
    {'Open': 'mean', 'High': 'mean', 'Low': 'mean', 'Close': 'mean', 'Adj Close': 'mean', 'Volume': 'sum'}
)

# Decompose the time series
decomposition = seasonal_decompose(df_weekly['Close'], model='additive', period=1)  # Adjust period
if needed

# Extract components
trend = decomposition.trend
seasonal = decomposition.seasonal
```

```
residual = decomposition.resid

# Plot the components
fig, axes = plt.subplots(4, 1, figsize=(14, 12))
axes[0].plot(df_weekly.index, df_weekly['Close'])  # Original data
axes[1].plot(df_weekly.index, trend)  # Trend
axes[2].plot(df_weekly.index, seasonal)  # Seasonality
axes[3].plot(df_weekly.index, residual)  # Residuals
plt.tight_layout()
plt.show()
```

OUTPUT:



RESULT:

Thus, the program using the time series data implementation has been done successfully.