7.Implement program forDecomposing Time Series Data into Trend and Seasonality

**Aim:**

The aim of this program is to decompose a given time series dataset into its trend and seasonal components. This is useful for analyzing the underlying patterns in the data, separating long-term trends from seasonal variations, and making more accurate predictions.

**Procedure:**

1. **Import Required Libraries:**  
   We need libraries like Pandas for data manipulation, Matplotlib for visualization, and statsmodels for time series decomposition.
2. **Load Data:**  
   Import the time series data (real or synthetic data).
3. **Decompose Time Series:**  
   Use statistical methods to decompose the time series into its components (trend, seasonal, and residual). The statsmodels.tsa.seasonal\_decompose() function can be used for this purpose.
4. **Visualize the Results:**  
   Plot the decomposed components: trend, seasonality, and residuals.
5. **Interpret the Results:**  
   Analyze the decomposed components to understand the underlying trends and seasonal patterns in the data.

**Code:**

Here is the Python code to decompose a time series into its trend and seasonal components:

python

CopyEdit

# Step 1: Import necessary libraries

import numpy as np

import pandas as pd

import matplotlib.pyplot as plt

from statsmodels.tsa.seasonal import seasonal\_decompose

# Step 2: Load or simulate time series data

# Example: Simulating some monthly data with seasonality and trend

data = pd.Series([100, 110, 120, 130, 140, 150, 160, 170, 180, 190, 200, 210,

220, 230, 240, 250, 260, 270, 280, 290, 300, 310, 320, 330],

index=pd.date\_range(start='2022-01-01', periods=24, freq='M'))

# Step 3: Decompose the time series

decomposition = seasonal\_decompose(data, model='additive', period=12)

# Step 4: Visualize the results

plt.figure(figsize=(10, 8))

plt.subplot(411)

plt.plot(data, label='Original Data')

plt.title('Original Time Series')

plt.subplot(412)

plt.plot(decomposition.trend, label='Trend Component', color='orange')

plt.title('Trend Component')

plt.subplot(413)

plt.plot(decomposition.seasonal, label='Seasonal Component', color='green')

plt.title('Seasonal Component')

plt.subplot(414)

plt.plot(decomposition.resid, label='Residual Component', color='red')

plt.title('Residual Component')

plt.tight\_layout()

plt.show()

# Step 5: Interpret the Results

print("Trend Component:\n", decomposition.trend.dropna())

print("Seasonal Component:\n", decomposition.seasonal)

print("Residual Component:\n", decomposition.resid.dropna())

**Explanation of the Code:**

1. **Data Simulation**:  
   The time series data is simulated using pd.Series with a trend and seasonal pattern. The data is indexed by monthly dates starting from January 2022.
2. **Decomposition**:  
   The seasonal\_decompose() function from the statsmodels library is used to decompose the time series into its trend, seasonal, and residual components. We use an "additive" model, which assumes that the components are added together.
3. **Visualization**:  
   The components (original, trend, seasonal, and residual) are plotted in four subplots for better comparison. Each component is displayed separately to understand how they contribute to the overall time series.
4. **Interpretation**:  
   After decomposition, the individual components are printed for further inspection. These can be analyzed to understand the underlying behavior of the time series.

**Result:**

After running the program, the following results will be displayed:

1. **Plots**:  
   Four subplots will be displayed:
   * **Original Time Series**: The raw data.
   * **Trend Component**: The long-term trend that represents the general direction of the data.
   * **Seasonal Component**: The repeating seasonal pattern or cycle present in the data.
   * **Residual Component**: The noise or random fluctuations left after removing the trend and seasonality.
2. **Printed Output**:  
   The components will be printed for further inspection:
   * **Trend Component**: Shows the underlying trend, which is the smoothed version of the data, removing seasonal effects.
   * **Seasonal Component**: Shows the seasonal variation, capturing periodic patterns over time.
   * **Residual Component**: Shows the remaining noise or random variation after trend and seasonality have been removed.