In [1]:
```
import pandas as pd
import matplotlib.pyplot as plt
from statsmodels.tsa.statespace.sarimax import SARIMAX
from sklearn.metrics import mean_absolute_error, mean_squared_error
import numpy as np
```

In [2]:
```
df = pd.read_csv(r"C:\Users\HP\OneDrive\Desktop\NLP\Data\ML471_S4_Datafile_Pract
```

In [3]:
```
df['Date'] = pd.to_datetime(df['Date'])
df.set_index('Date', inplace=True)
```

In [4]:
```
close_series = df['Close']
```

In [5]:
```
train_size = int(len(close_series) * 0.8)
train = close_series[:train_size]
test = close_series[train_size:]
```

In [6]:
```
model = SARIMAX(
    train,
    order=(1, 0, 1),
    seasonal_order=(3, 1, 1, 12),
    enforce_stationarity=False,
    enforce_invertibility=False
)

model_fit = model.fit()
```

```
c:\Users\HP\AppData\Local\Programs\Python\Python312\Lib\site-packages\statsmodels
\tsa\base\tsa_model.py:473: ValueWarning: No frequency information was provided,
so inferred frequency ME will be used.
  self._init_dates(dates, freq)
c:\Users\HP\AppData\Local\Programs\Python\Python312\Lib\site-packages\statsmodels
\tsa\base\tsa_model.py:473: ValueWarning: No frequency information was provided,
so inferred frequency ME will be used.
  self._init_dates(dates, freq)
```

In [7]:
```
forecast = model_fit.forecast(steps=len(test))
```

In [8]:
```
mae = mean_absolute_error(test, forecast)
rmse = np.sqrt(mean_squared_error(test, forecast))

print("MAE:", mae)
print("RMSE:", rmse)
```

```
MAE: 10.640486346315312
RMSE: 13.785222227411309
```

In [9]:
```
plt.figure(figsize=(12,6))
plt.plot(train, label='Train', color='blue')
plt.plot(test, label='Actual', color='orange', linestyle='--')
plt.plot(test.index, forecast, label='Forecast', color='green', linestyle='--')

plt.title("SARIMA((1, 0, 1)x(3, 1, 1, 12) Forecast Plot")
plt.xlabel("Date")
plt.ylabel("Close Price")
plt.legend()
```

```
plt.grid(True)
plt.show()
```