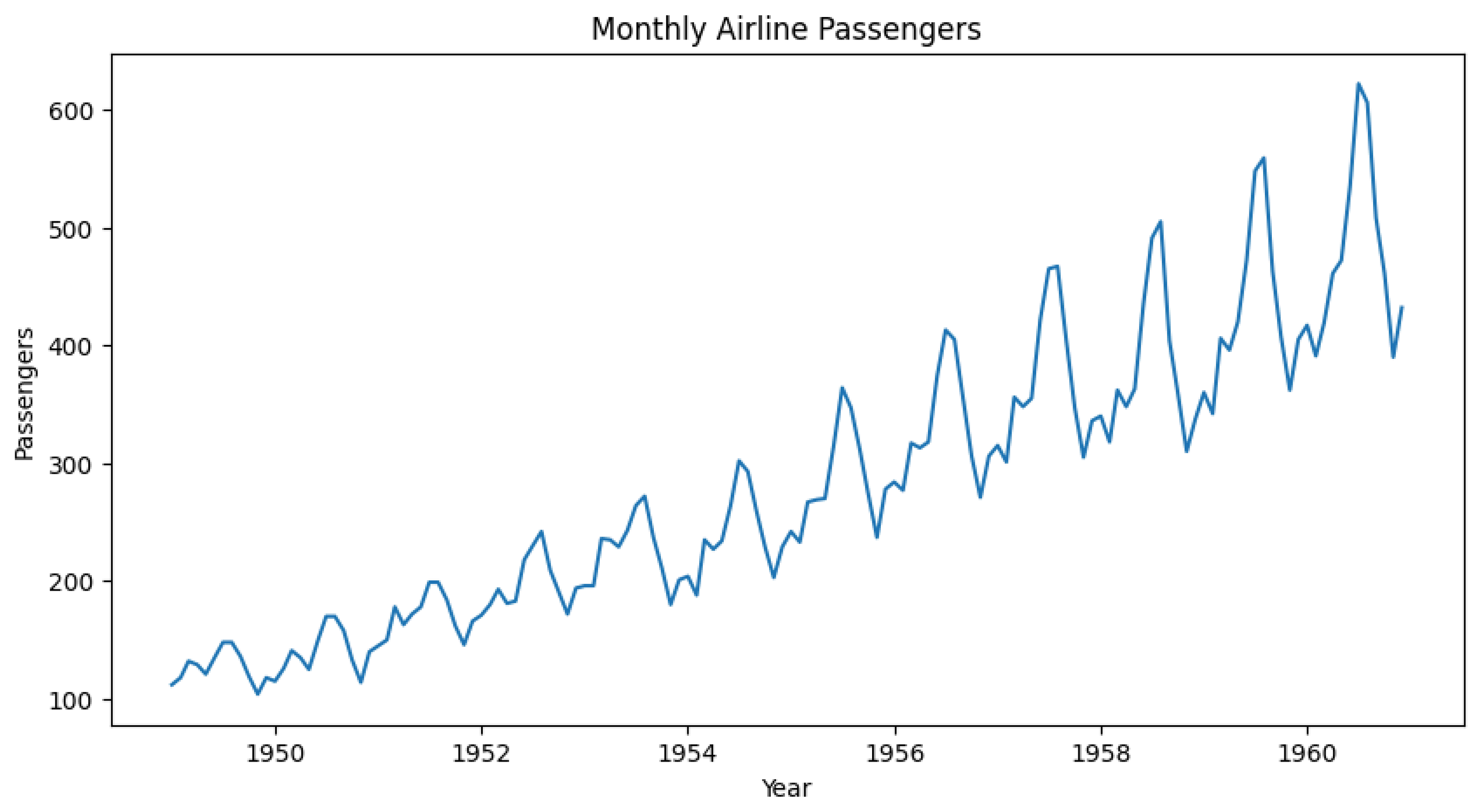
In [1]:
```
import pandas as pd
import numpy as np
import matplotlib.pyplot as plt

from statsmodels.tsa.stattools import adfuller
from statsmodels.graphics.tsaplots import plot_acf, plot_pacf
from statsmodels.tsa.arima.model import ARIMA
```

In [2]:
```
path = r"C:\Users\Mathusri O\Downloads\archive\AirPassengers.csv"
df = pd.read_csv(path)

df['Month'] = pd.to_datetime(df['Month'])
df.set_index('Month', inplace=True)

print(df.head())
```

```
            #Passengers
Month
1949-01-01          112
1949-02-01          118
1949-03-01          132
1949-04-01          129
1949-05-01          121
```

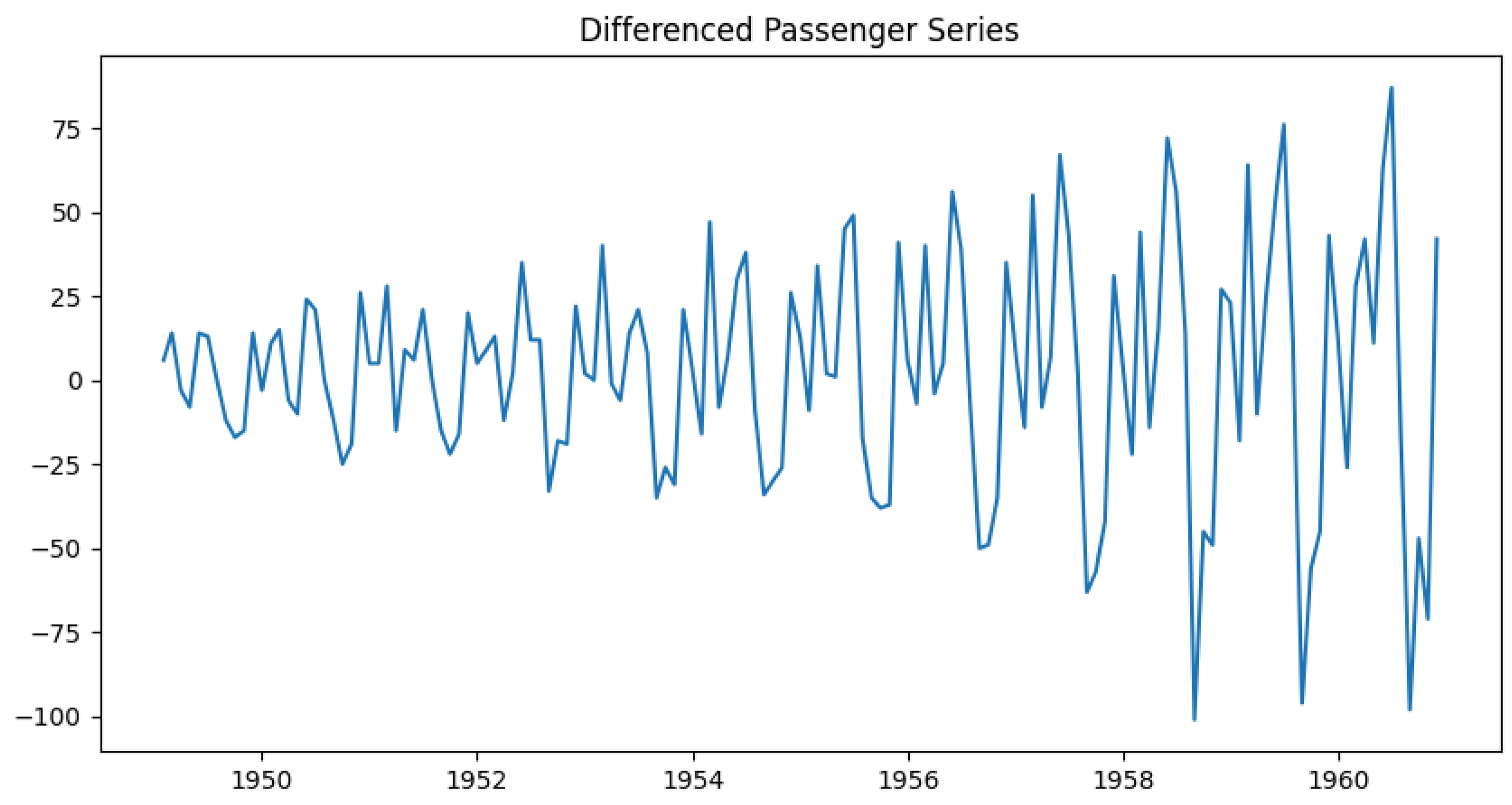
In [4]:
```
plt.figure(figsize=(10,5))
plt.plot(df['#Passengers'])
plt.title("Monthly Airline Passengers")
plt.xlabel("Year")
plt.ylabel("Passengers")
plt.show()
```



In [6]:
```
def adf_test(series):
    result = adfuller(series)
    print("ADF Statistic:", result[0])
    print("p-value:", result[1])

adf_test(df['#Passengers'])
```

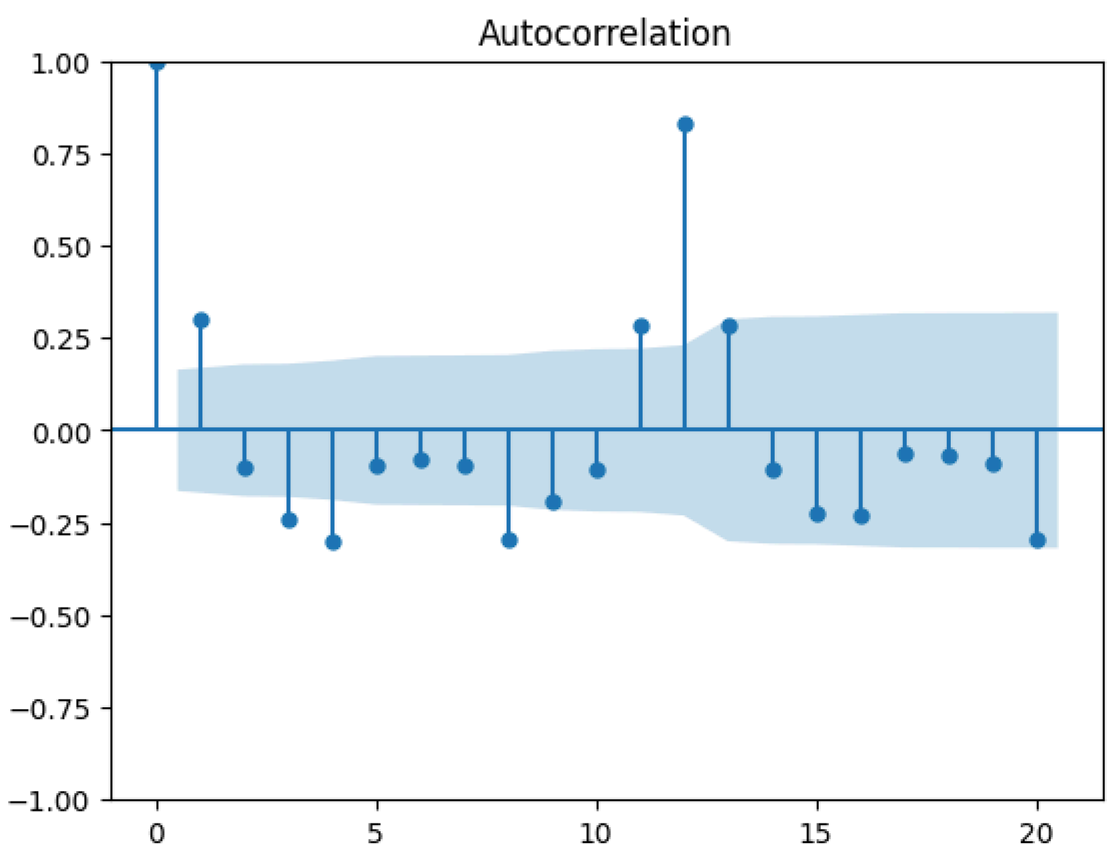
```
ADF Statistic: 0.8153688792060482
p-value: 0.991880243437641
```

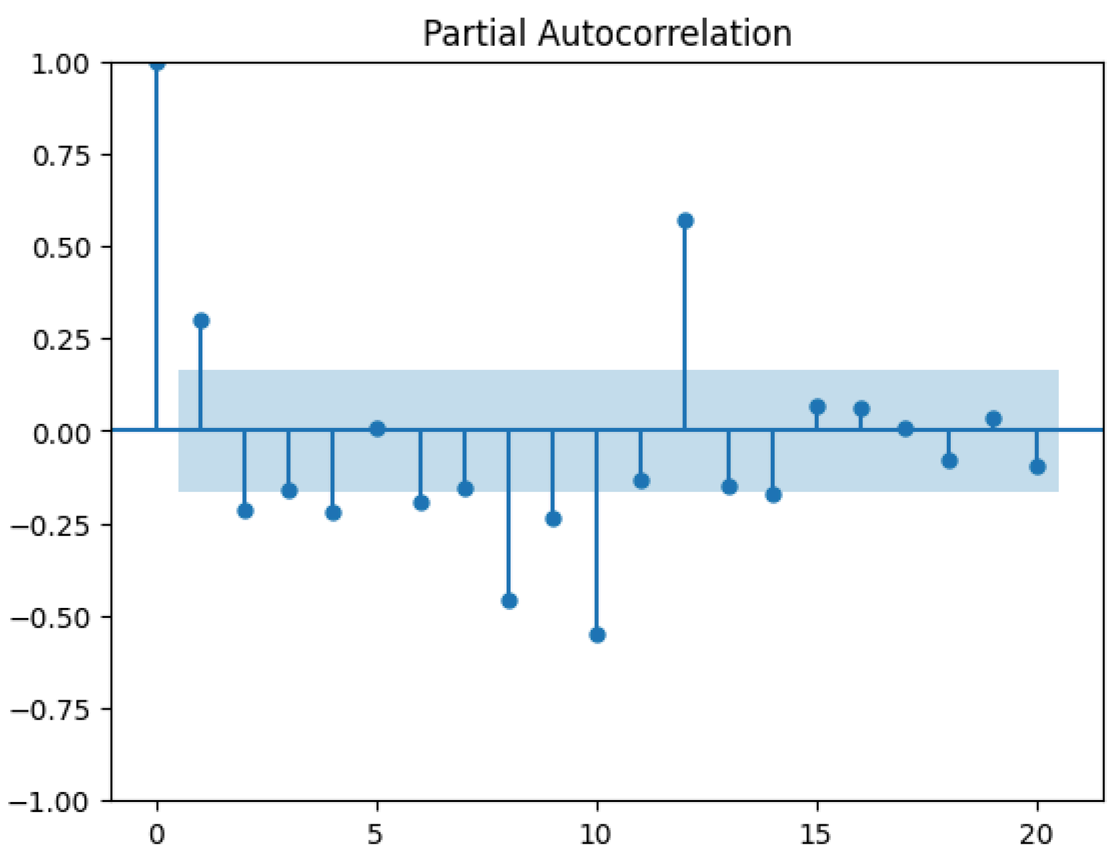
In [8]:
```
df['Passengers_diff'] = df['#Passengers'].diff().dropna()

plt.figure(figsize=(10,5))
plt.plot(df['Passengers_diff'])
plt.title("Differenced Passenger Series")
plt.show()

adf_test(df['Passengers_diff'].dropna())
```




```
ADF Statistic: -2.8292668241699994
p-value: 0.0542132902838255
```

In [9]:
```
plot_acf(df['Passengers_diff'].dropna(), lags=20)
plt.show()

plot_pacf(df['Passengers_diff'].dropna(), lags=20)
plt.show()
```

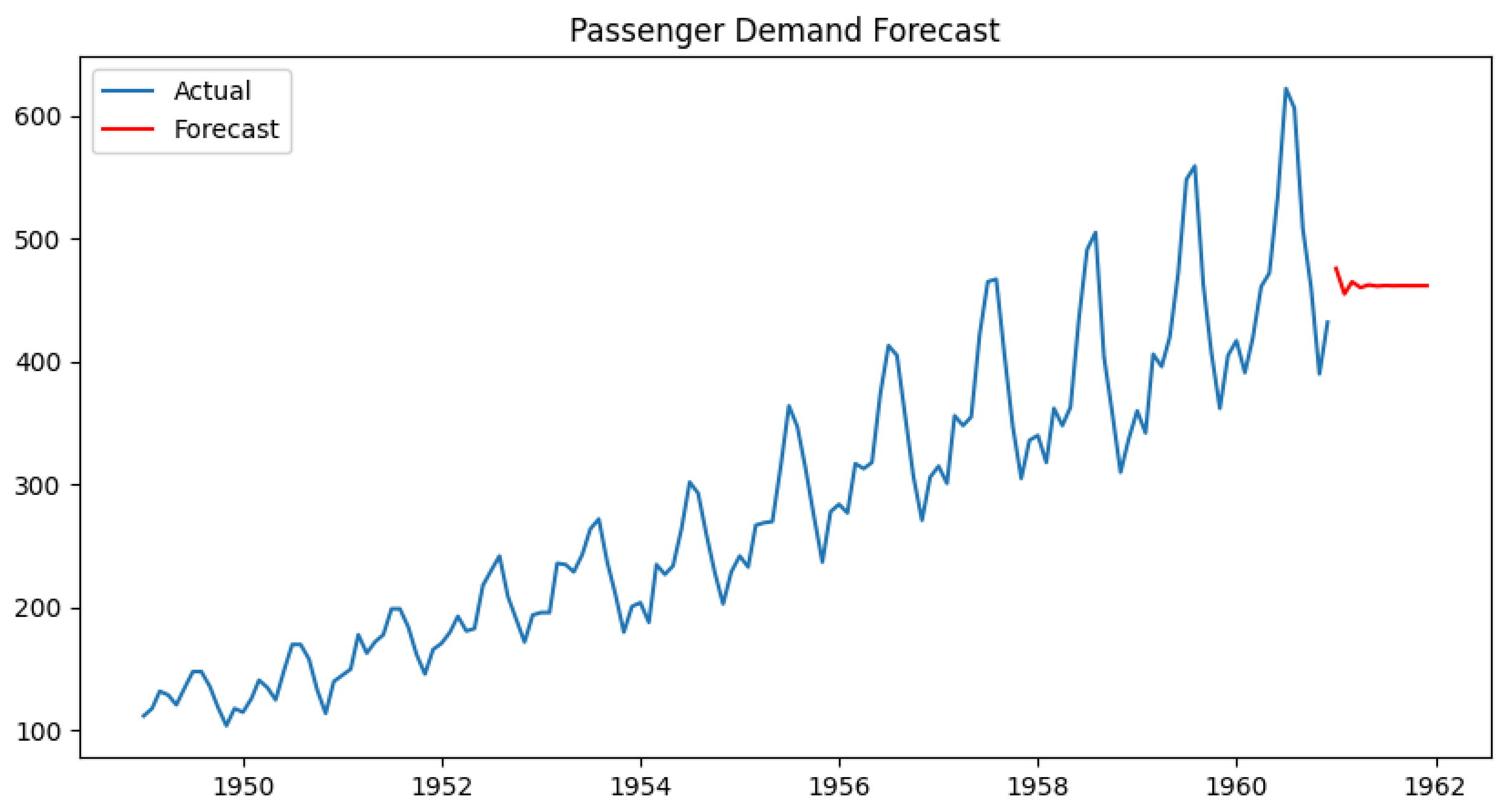




In [11]:
```
model = ARIMA(df['#Passengers'], order=(1,1,1))
model_fit = model.fit()

print(model_fit.summary())
```

```
C:\Users\Mathusri O\AppData\Roaming\Python\Python310\site-packages\statsmodels\tsa\base\tsa_model.py:473: ValueWarning: No frequency information was provided, so inferred frequency MS will be used.
  self._init_dates(dates, freq)
C:\Users\Mathusri O\AppData\Roaming\Python\Python310\site-packages\statsmodels\tsa\base\tsa_model.py:473: ValueWarning: No frequency information was provided, so inferred frequency MS will be used.
  self._init_dates(dates, freq)
C:\Users\Mathusri O\AppData\Roaming\Python\Python310\site-packages\statsmodels\tsa\base\tsa_model.py:473: ValueWarning: No frequency information was provided, so inferred frequency MS will be used.
  self._init_dates(dates, freq)
                               SARIMAX Results
==============================================================================
Dep. Variable:            #Passengers   No. Observations:                  144
Model:                 ARIMA(1, 1, 1)   Log Likelihood                -694.341
Date:                Sat, 07 Feb 2026   AIC                           1394.683
Time:                        15:21:22   BIC                           1403.571
Sample:                    01-01-1949   HQIC                          1398.294
                         - 12-01-1960
Covariance Type:                  opg
==============================================================================
                 coef    std err          z      P>|z|      [0.025      0.975]
------------------------------------------------------------------------------
ar.L1         -0.4742      0.123     -3.847      0.000      -0.716      -0.233
ma.L1          0.8635      0.078     11.051      0.000       0.710       1.017
sigma2       961.9270    107.433      8.954      0.000     751.362    1172.492
===================================================================================
Ljung-Box (L1) (Q):                   0.21   Jarque-Bera (JB):                 2.14
Prob(Q):                              0.65   Prob(JB):                         0.34
Heteroskedasticity (H):               7.00   Skew:                            -0.21
Prob(H) (two-sided):                  0.00   Kurtosis:                         3.43
===================================================================================

Warnings:
[1] Covariance matrix calculated using the outer product of gradients (complex-step).
```

In [13]:
```
forecast = model_fit.forecast(steps=12)

plt.figure(figsize=(10,5))
plt.plot(df['#Passengers'], label="Actual")
plt.plot(forecast, label="Forecast", color='red')
plt.legend()
plt.title("Passenger Demand Forecast")
plt.show()

print(forecast)
```




```
1961-01-01    475.735059
1961-02-01    454.996073
1961-03-01    464.830415
1961-04-01    460.167010
1961-05-01    462.378378
1961-06-01    461.329756
1961-07-01    461.827008
1961-08-01    461.591213
1961-09-01    461.703026
1961-10-01    461.650005
1961-11-01    461.675148
1961-12-01    461.663225
Freq: MS, Name: predicted_mean, dtype: float64
```

In [ ]: