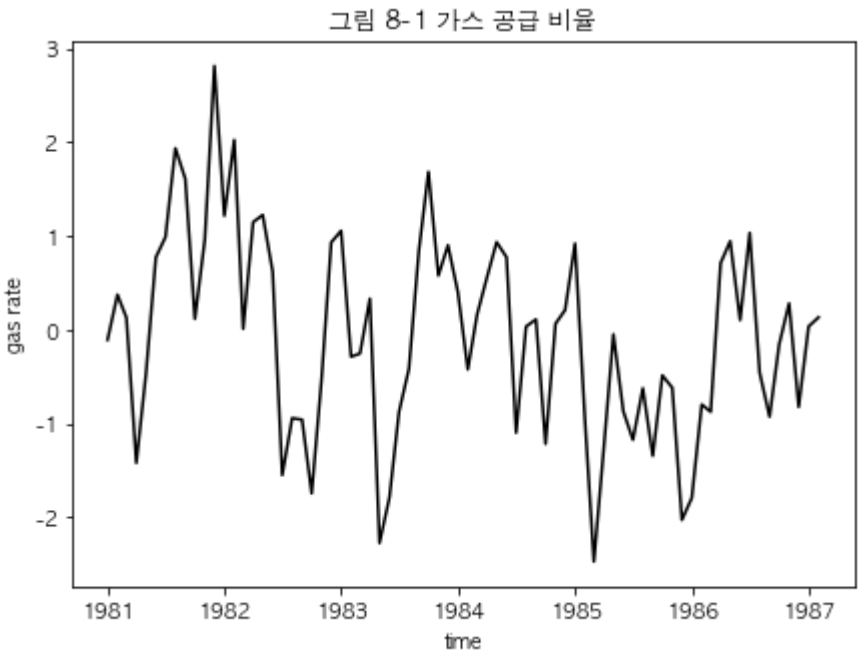
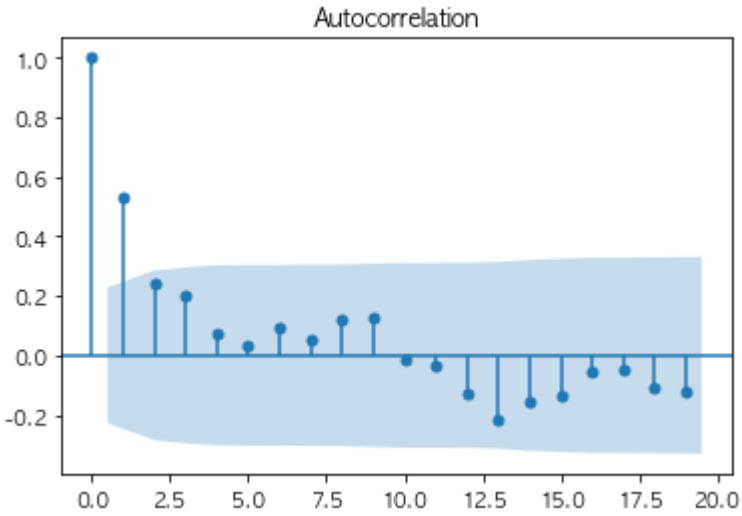
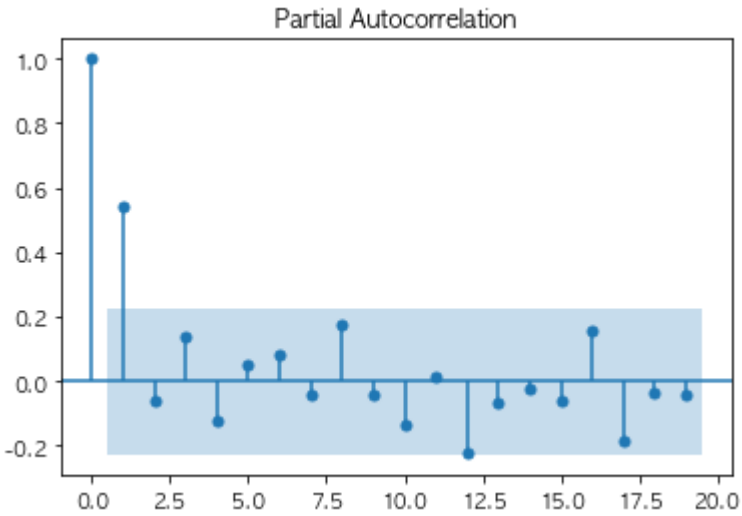
In [7]:
```
import math
import numpy as np
import pandas as pd
import matplotlib.pyplot as plt
plt.rc('font', family='AppleGothic')
plt.rcParams['axes.unicode_minus'] = False

from statsmodels.tsa.arima_model import ARIMA
from statsmodels.graphics.tsaplots import plot_acf, plot_pacf
```

In [6]:
```
# Example 4.1
z = []

temp = []
with open('../data/gas.txt') as f:
    for line in f.readlines():
        for elem in line.rstrip().split(" "):
            if len(elem) and len(temp) < 2:
                temp.append(float(elem))

        if len(temp) >= 2:
            z.append(temp)
            temp = []

index = pd.date_range(start="1981", periods=len(z), freq="MS")
data = pd.DataFrame(z, index=index, columns=["rate", "co2"])

fig, ax = plt.subplots(figsize=(7, 5))
ax.plot(data['rate'], 'black')
ax.set_xlabel("time")
ax.set_ylabel("gas rate")
ax.set_title("그림 8-1 가스 공급 비율")
plt.show()
```



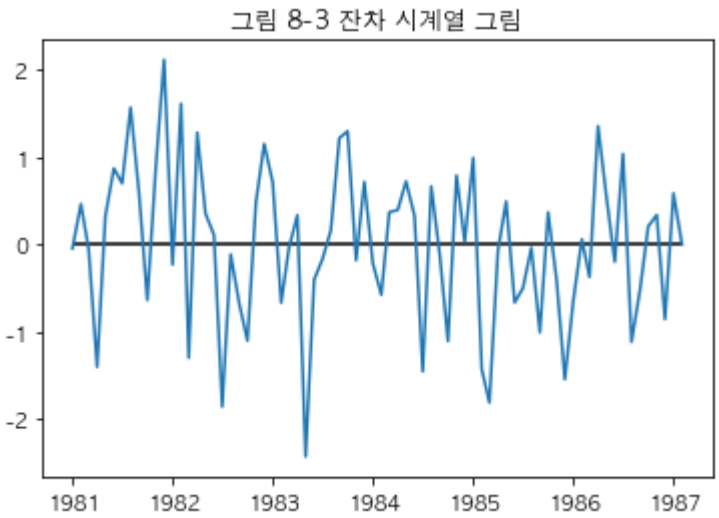
In [10]:
```
plot_acf(data['rate'])
plot_pacf(data['rate'])
plt.show()
```



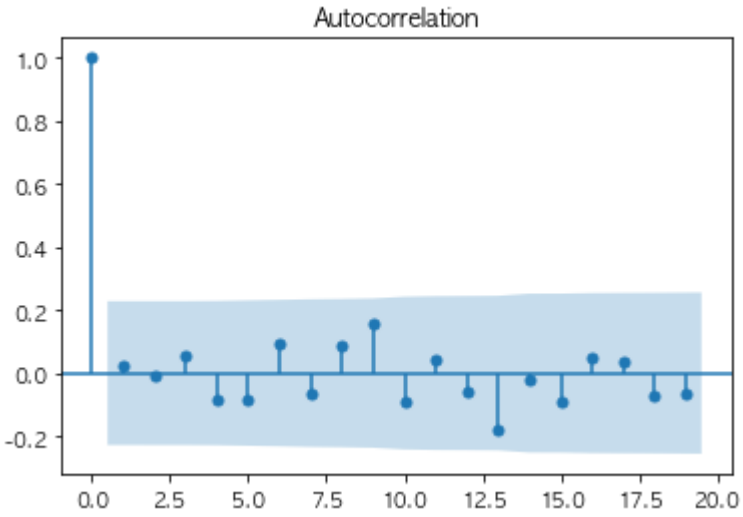


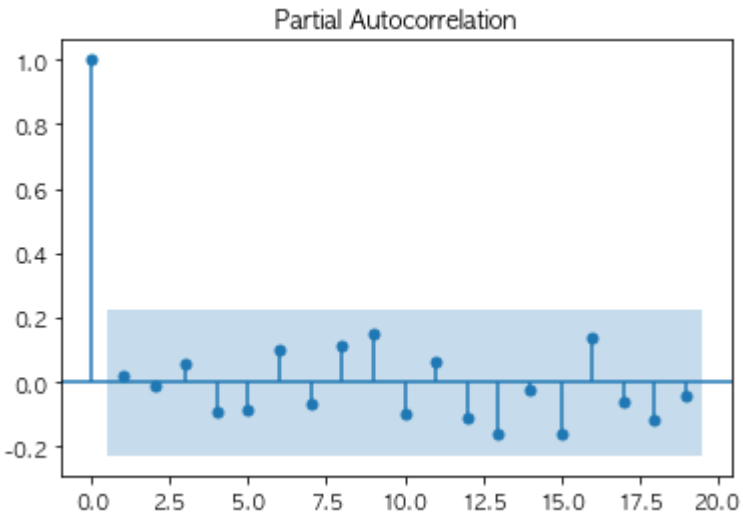
In [24]:
```
model = ARIMA(data['rate'], order=(3, 0, 0)).fit()
resid = model.resid

plt.plot(resid)
plt.title("그림 8-3 잔차 시계열 그림")
plt.hlines(0, resid.index.min(), resid.index.max(), color="black")
plt.show()
```

```
RUNNING THE L-BFGS-B CODE

           * * *

Machine precision = 2.220D-16
 N =            4     M =           12

At X0         0 variables are exactly at the bounds

At iterate    0    f=  1.30721D+00    |proj g|=  3.85114D-03

At iterate    5    f=  1.30715D+00    |proj g|=  1.77636D-07

           * * *

Tit   = total number of iterations
Tnf   = total number of function evaluations
Tnint = total number of segments explored during Cauchy searches
Skip  = number of BFGS updates skipped
Nact  = number of active bounds at final generalized Cauchy point
Projg = norm of the final projected gradient
F     = final function value

           * * *

   N    Tit     Tnf  Tnint  Skip  Nact     Projg        F
    4      7      9      1     0     0   2.220D-08   1.307D+00
  F =   1.3071460988809354

CONVERGENCE: REL_REDUCTION_OF_F_<=_FACTR*EPSMCH

 This problem is unconstrained.
```

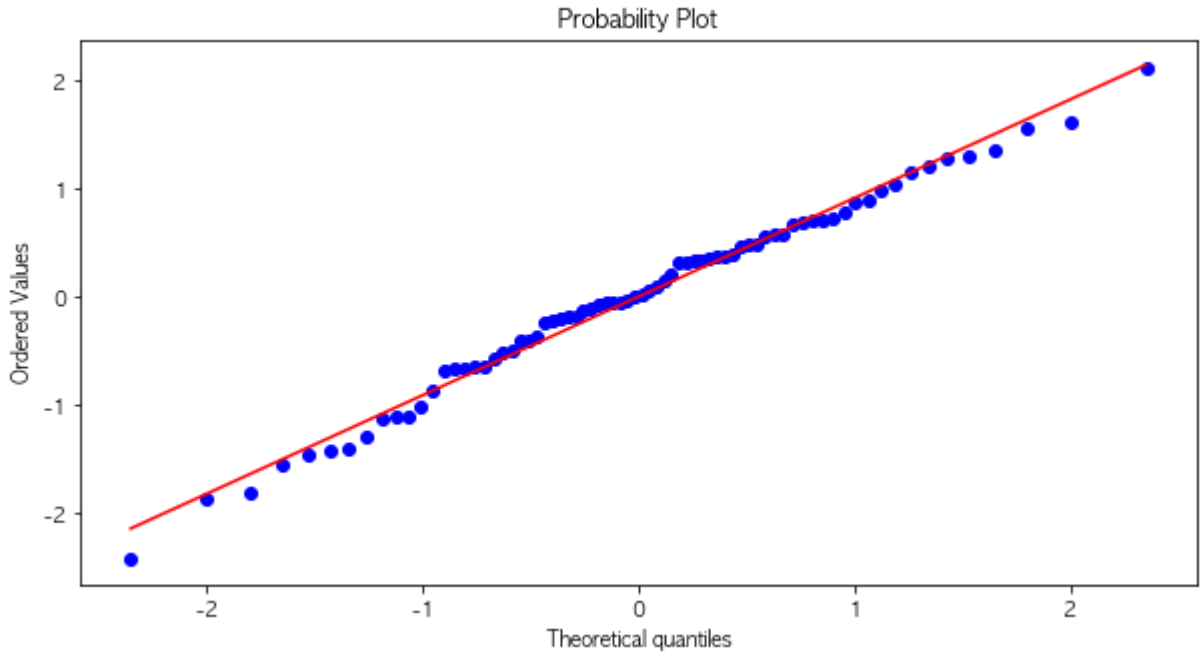


In [25]:
```
plot_acf(resid)
plot_pacf(resid)
plt.show()
```

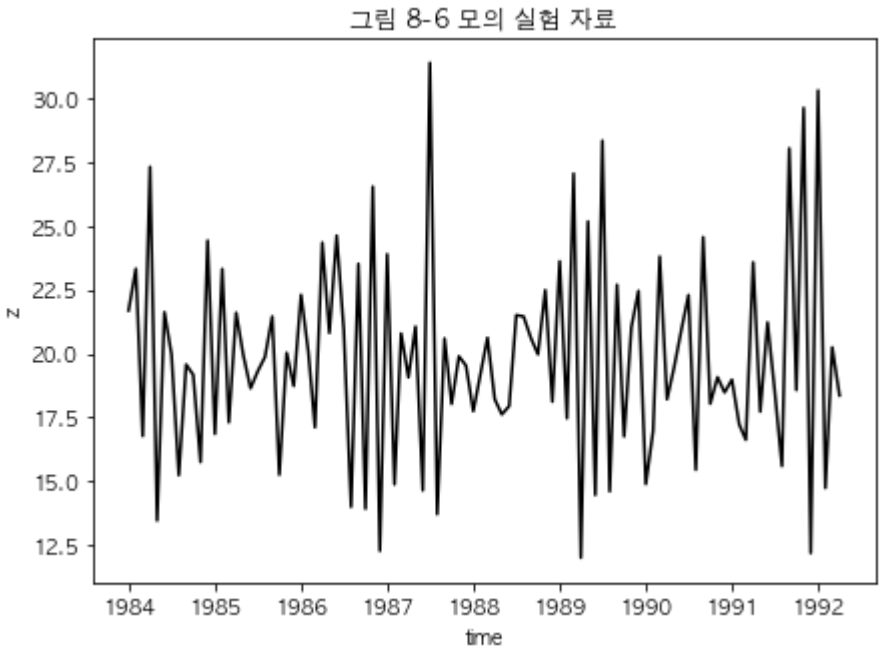




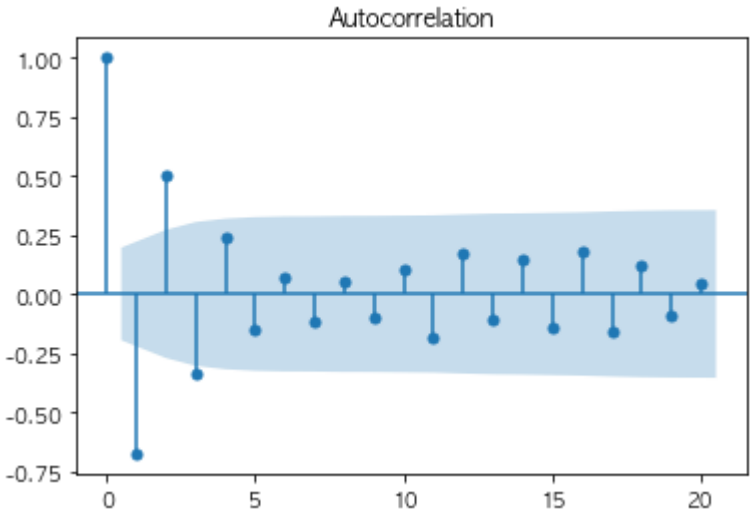
In [27]:
```
import scipy.stats as stats

plt.figure(figsize=(10, 5))
plt.title("그림 8-5 잔차의 정규성 검정")
stats.probplot(resid, dist=stats.norm, plot=plt)
plt.show()
```

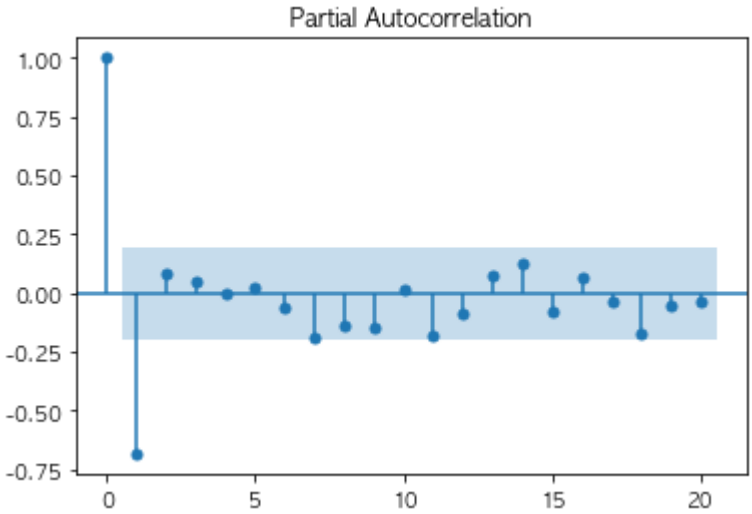


In [28]:
```
# Example 8-7
z = []

with open('../data/eg8_7.txt') as f:
    for line in f.readlines():
        for elem in line.rstrip().split(" "):
            if len(elem):
                z.append(float(elem))

index = pd.date_range(start="1984", periods=len(z), freq="MS")
data = pd.Series(z, index)

fig, ax = plt.subplots(figsize=(7, 5))
ax.plot(data, 'black')
ax.set_xlabel("time")
ax.set_ylabel("z")
ax.set_title("그림 8-6 모의 실험 자료")
plt.show()
```



In [30]:
```
plot_acf(data)
plot_pacf(data)
plt.show()
```

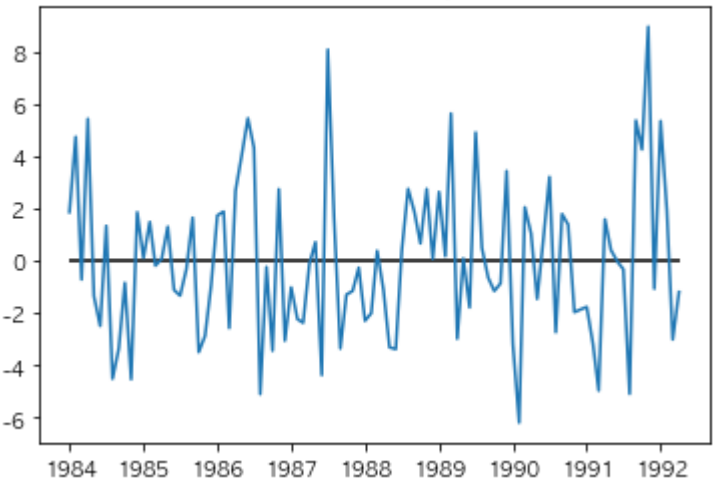




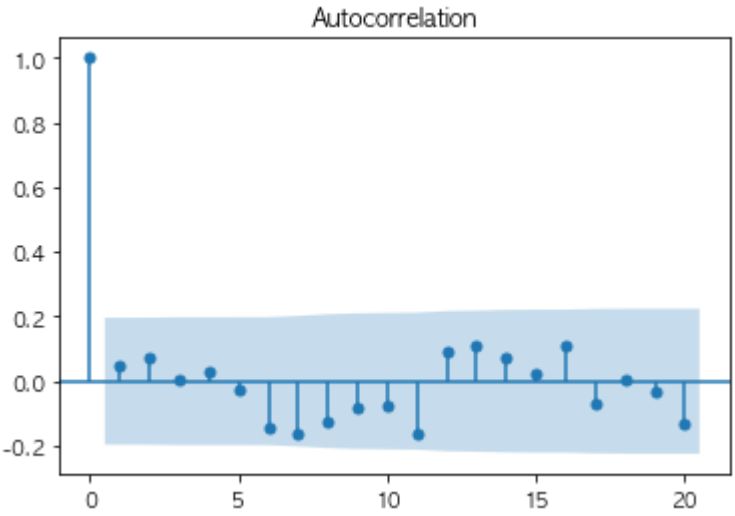
In [31]:
```
model = ARIMA(data, order=(1, 0, 0)).fit()
resid = model.resid

plt.plot(resid)
plt.title("그림 8-7 잔차 시계열 그림")
plt.hlines(0, resid.index.min(), resid.index.max(), color="black")
plt.show()
```

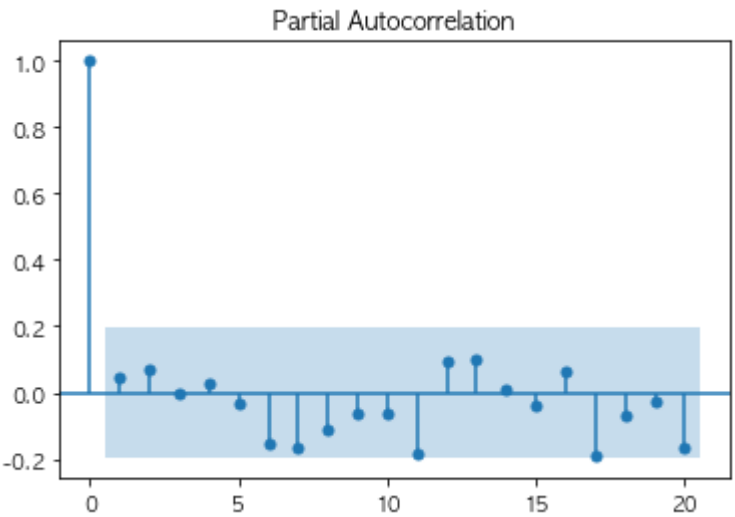
```
RUNNING THE L-BFGS-B CODE

           * * *

Machine precision = 2.220D-16
 N =            2     M =           12

At X0         0 variables are exactly at the bounds

At iterate    0    f=  2.50614D+00    |proj g|=  2.65561D-03

At iterate    5    f=  2.50611D+00    |proj g|=  4.44089D-08

           * * *

Tit   = total number of iterations
Tnf   = total number of function evaluations
Tnint = total number of segments explored during Cauchy searches
Skip  = number of BFGS updates skipped
Nact  = number of active bounds at final generalized Cauchy point
Projg = norm of the final projected gradient
F     = final function value

           * * *

   N    Tit     Tnf  Tnint  Skip  Nact     Projg        F
    2      5      7      1     0     0   4.441D-08   2.506D+00
  F =   2.5061142051458765

CONVERGENCE: REL_REDUCTION_OF_F_<=_FACTR*EPSMCH

/Users/jonghyun/miniforge3/lib/python3.9/site-packages/statsmodels/tsa/arima_mo
del.py:472: FutureWarning:
statsmodels.tsa.arima_model.ARMA and statsmodels.tsa.arima_model.ARIMA have
been deprecated in favor of statsmodels.tsa.arima.model.ARIMA (note the .
between arima and model) and
statsmodels.tsa.SARIMAX. These will be removed after the 0.12 release.

statsmodels.tsa.arima.model.ARIMA makes use of the statespace framework and
is both well tested and maintained.

To silence this warning and continue using ARMA and ARIMA until they are
removed, use:

import warnings
warnings.filterwarnings('ignore', 'statsmodels.tsa.arima_model.ARMA',
                        FutureWarning)
warnings.filterwarnings('ignore', 'statsmodels.tsa.arima_model.ARIMA',
                        FutureWarning)

  warnings.warn(ARIMA_DEPRECATION_WARN, FutureWarning)
 This problem is unconstrained.
```

그림 8-7 잔차 시계열 그림

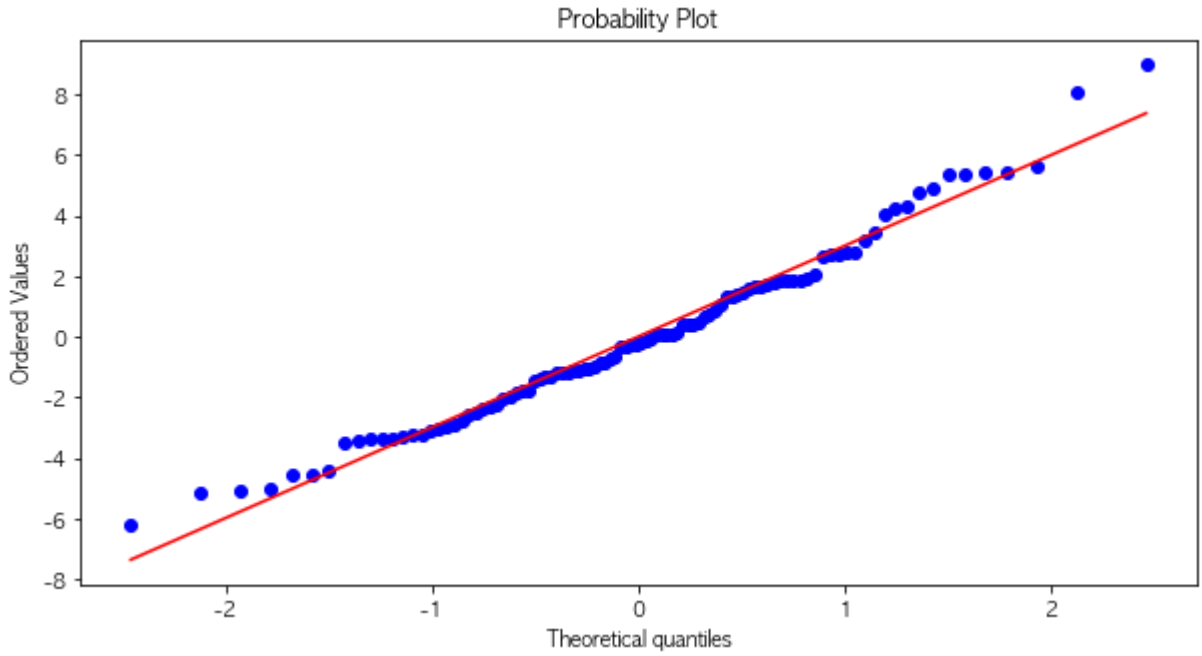


In [32]:
```
plot_acf(resid)
plot_pacf(resid)
plt.show()
```

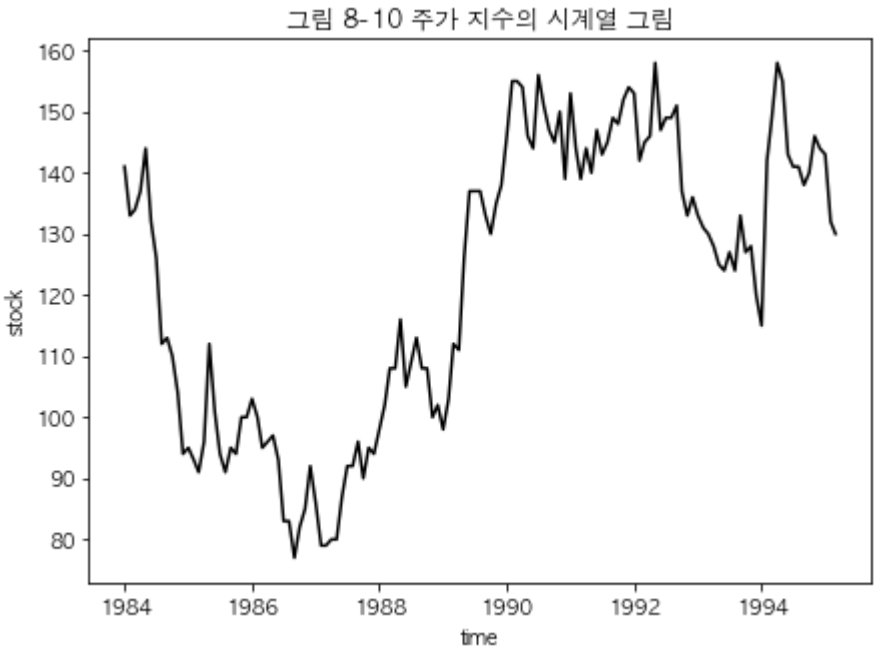




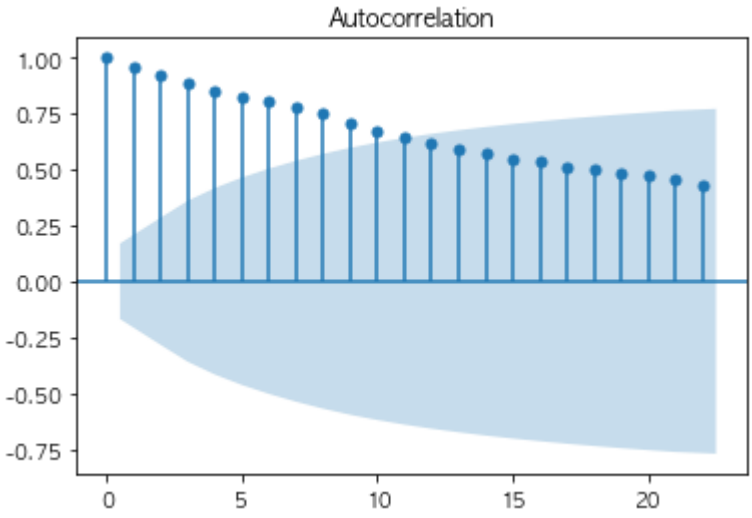
In [34]:
```
plt.figure(figsize=(10, 5))
plt.title("그림 8-9 잔차의 정규성 검정")
stats.probplot(resid, dist=stats.norm, plot=plt)
plt.show()
```

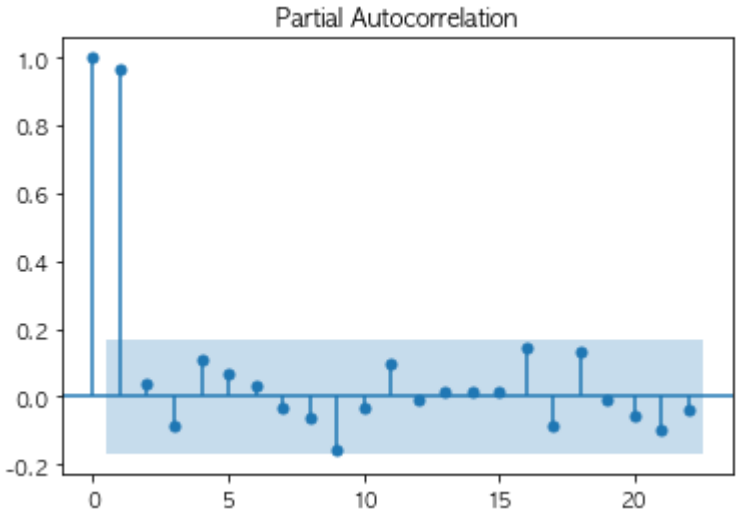


In [36]:
```
# Example 8-8
z = []

with open('../data/elecstock.txt') as f:
    for line in f.readlines():
        for elem in line.rstrip().split(" "):
            if len(elem):
                z.append(float(elem))

index = pd.date_range(start="1984", periods=len(z), freq="MS")
data = pd.Series(z, index)

fig, ax = plt.subplots(figsize=(7, 5))
ax.plot(data, 'black')
ax.set_xlabel("time")
ax.set_ylabel("stock")
ax.set_title("그림 8-10 주가 지수의 시계열 그림")
plt.show()
```

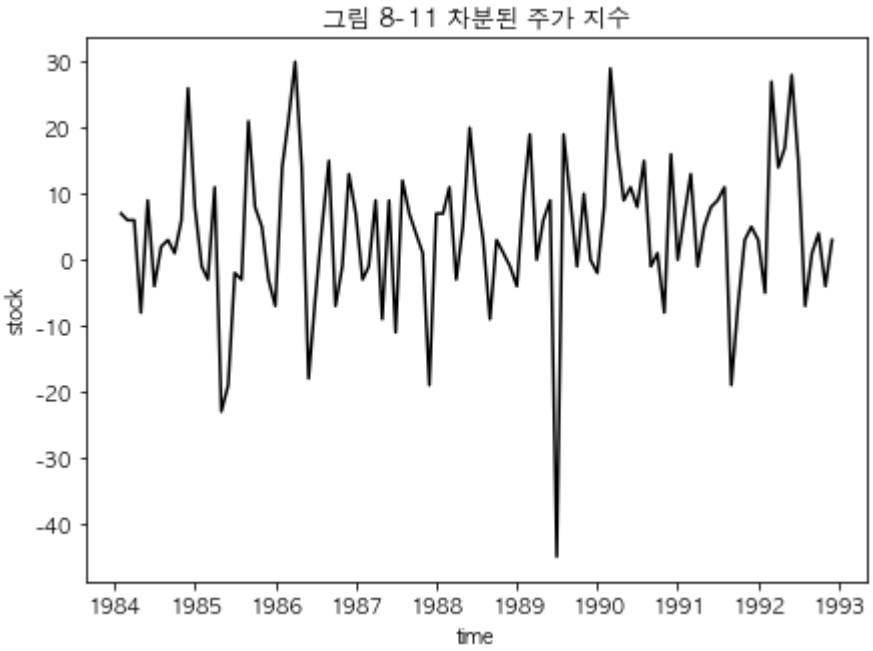


In [37]:
```
plot_acf(data)
plot_pacf(data)
plt.show()
```

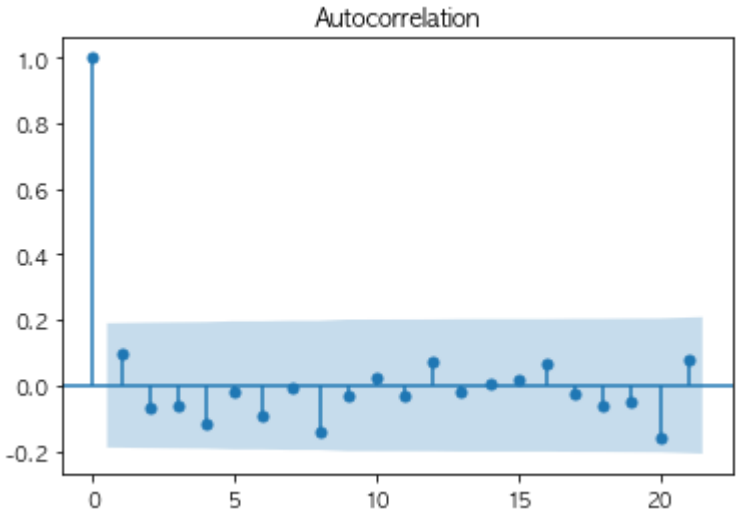


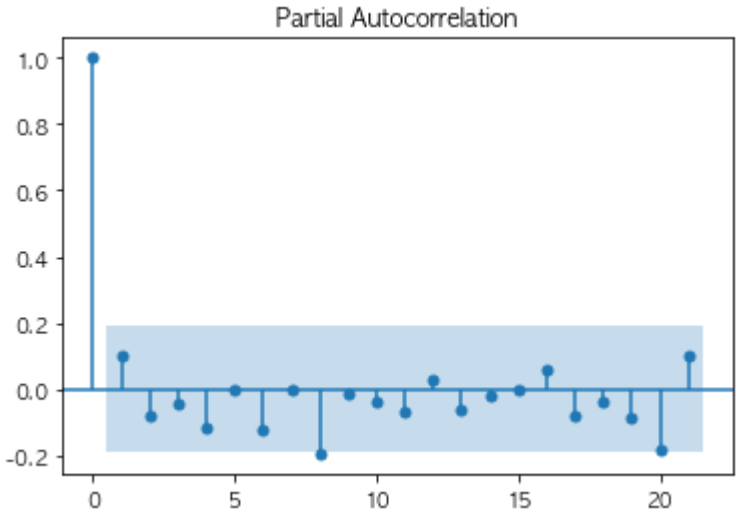


In [46]:
```
diff_data = data.diff(1)

fig, ax = plt.subplots(figsize=(7, 5))
ax.plot(diff_data, 'black')
ax.set_xlabel("time")
ax.set_ylabel("stock")
ax.hlines(0, diff_data.index.min(), diff_data.index.max())
ax.set_title("그림 8-11 차분된 주가 지수")
plt.show()
```



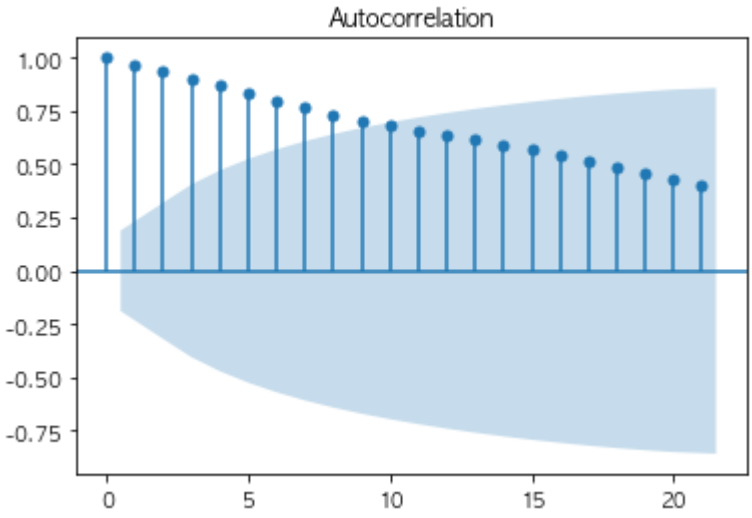
In [47]:
```
plot_acf(diff_data[1:])
plot_pacf(diff_data[1:])
plt.show()
```

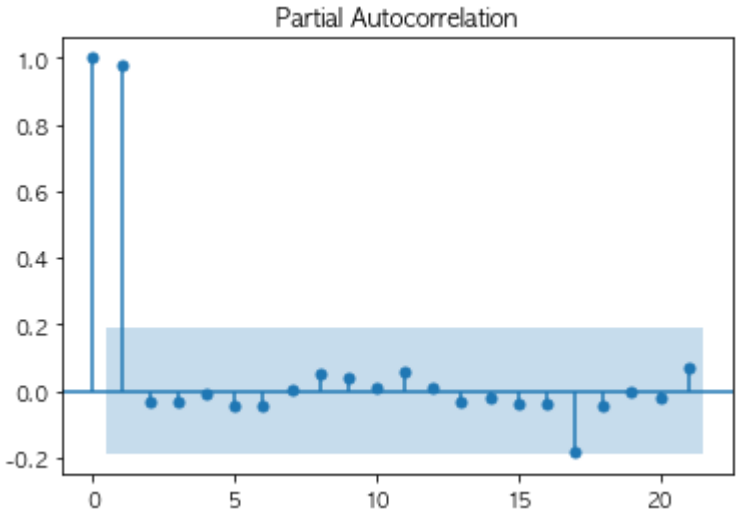




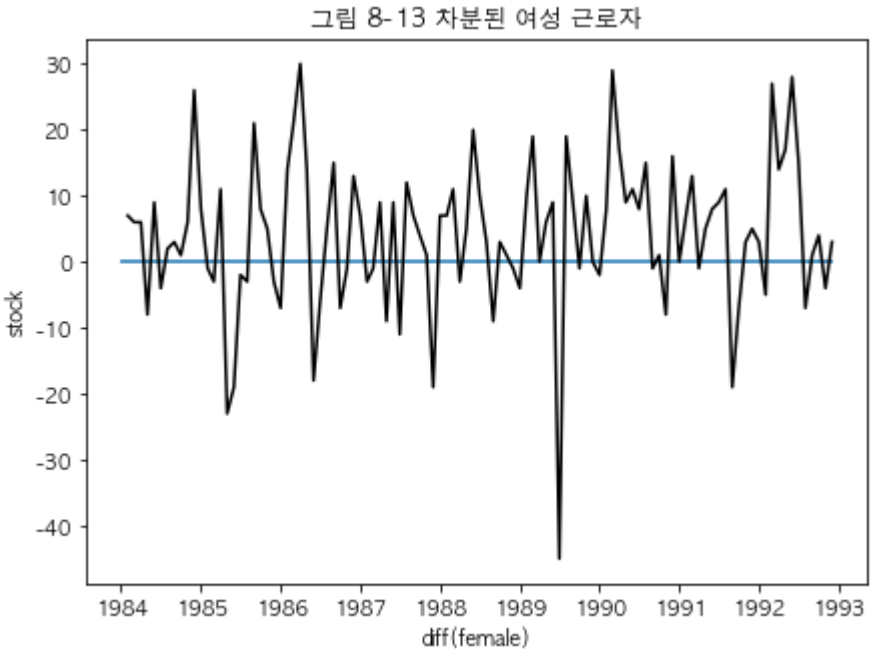
In [49]:
```
# Example 8-9
z = []

with open('../data/female.txt') as f:
    for line in f.readlines():
        for elem in line.rstrip().split(" "):
            if len(elem):
                z.append(float(elem))

index = pd.date_range(start="1984", periods=len(z), freq="MS")
data = pd.Series(z, index)

fig, ax = plt.subplots(figsize=(7, 5))
ax.plot(data, 'black')
ax.set_xlabel("time")
ax.set_ylabel("stock")
ax.set_title("그림 8-12 여성 근로자")
plt.show()
```

In [50]:
```
plot_acf(data)
plot_pacf(data)
plt.show()
```

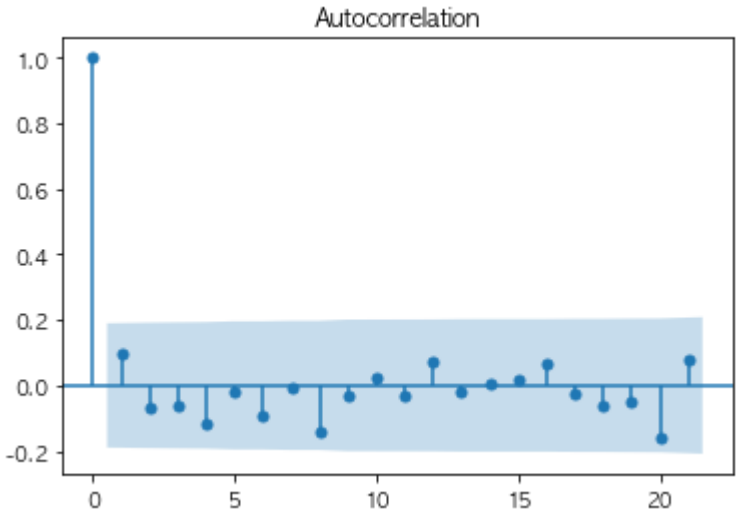


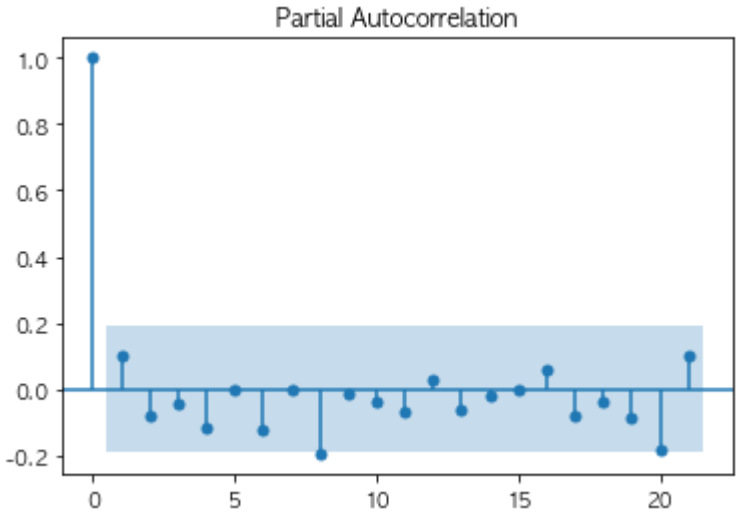


In [53]:
```
diff_data = data.diff(1)

fig, ax = plt.subplots(figsize=(7, 5))
ax.plot(diff_data, 'black')
ax.set_xlabel("diff(female)")
ax.set_ylabel("stock")
ax.hlines(0, diff_data.index.min(), diff_data.index.max())
ax.set_title("그림 8-13 차분된 여성 근로자")
plt.show()
```



In [55]:
```
plot_acf(diff_data[1:])
plot_pacf(diff_data[1:])
plt.show()
```





In [ ]: