Write a function that calculates the values of AR(p) model. The function must have a parameter burnin that determines how many initial values are discarded

In [21]:
```
import numpy as np
import pandas as pd
import math
import matplotlib.pyplot as plt
from random import gauss
from statsmodels.graphics.tsaplots import plot_acf, plot_pacf


def NextValue(p,prev_values):
    sum=0
    for i in np.arange(p):
        sum+=phis[i]*prev_values[len(prev_values)-p+i]
    return c+sum+np.random.randn()

def AR(p,c,phis):
    ARvalues=start_numbers
    for i in np.arange(simulation_length):
        ARvalues.append(NextValue(p,ARvalues))
    del ARvalues[0:burnin]
    return ARvalues
```

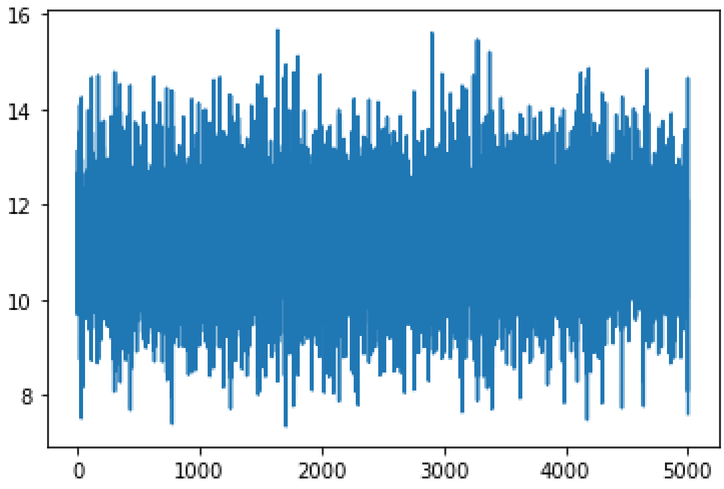
Calculate n = 5000 values of AR(1) model yt = 18 − 0.6yt−1 + ϵt.

In [27]:
```
p=1
c=18
phis=[-0.6]
burnin=300
start_numbers=[1,2,3]
simulation_length=5300



result=AR(p,c,phis)
plt.plot(pd.Series(result))
```

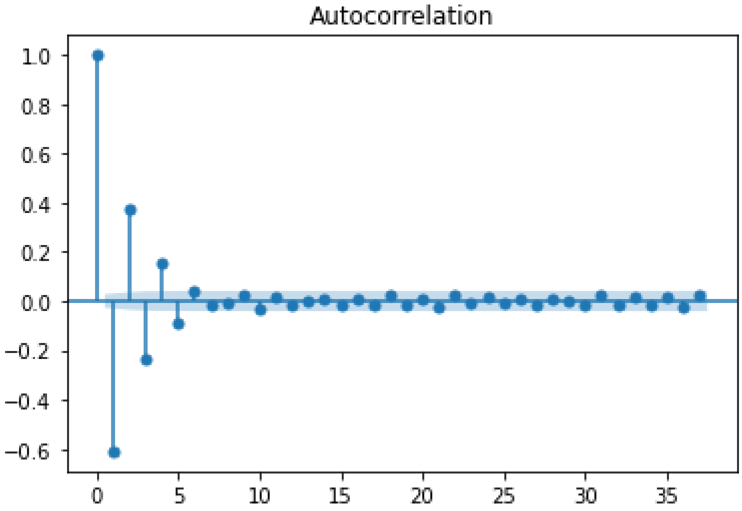
Out[27]:
```
[<matplotlib.lines.Line2D at 0x20086351610>]
```

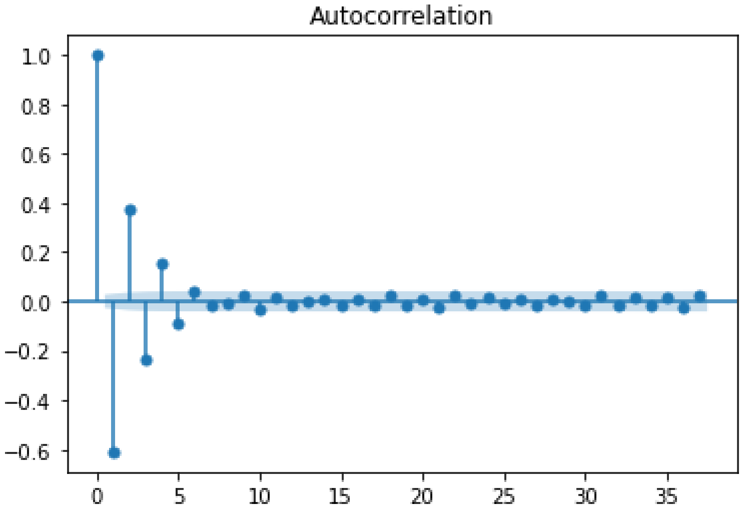


Calculate the autocorrelation (ACF) and partial autocorrelation (PACF) function for this time series

In [28]:
```
plot_acf(pd.Series(result))
```

Out[28]:

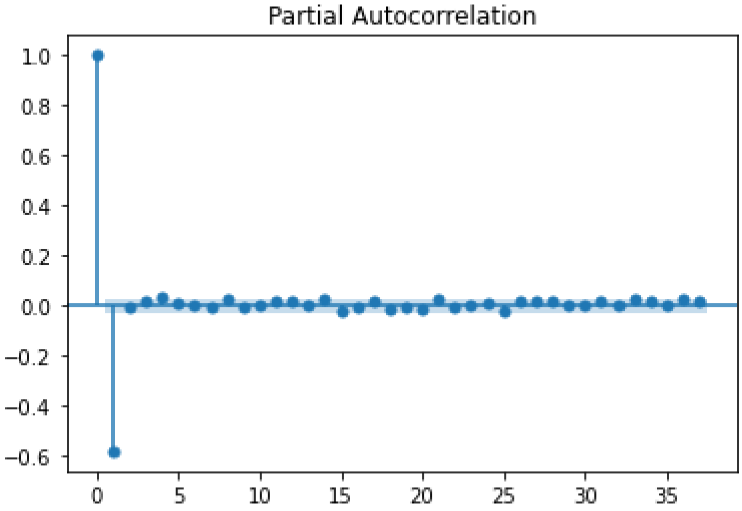


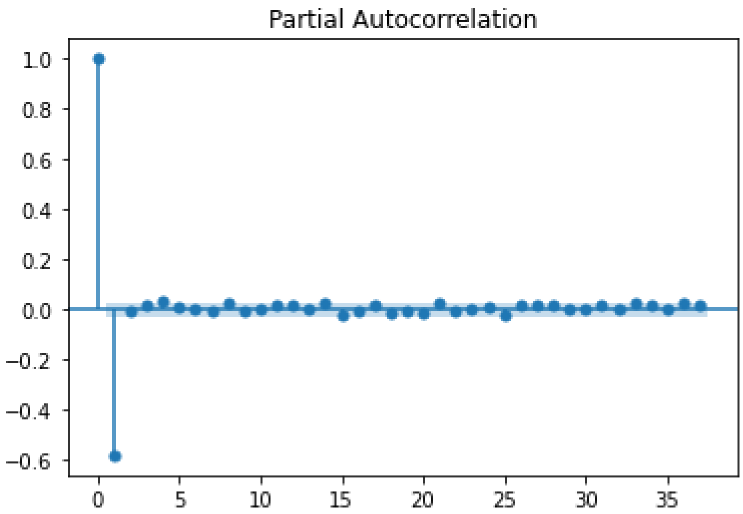


In [4]:

```
plot_pacf(pd.Series(result))
```

Out[4]:





Repeat the calculations for ϕ1 = −0.7, −0.8, −0.9.

In [30]:

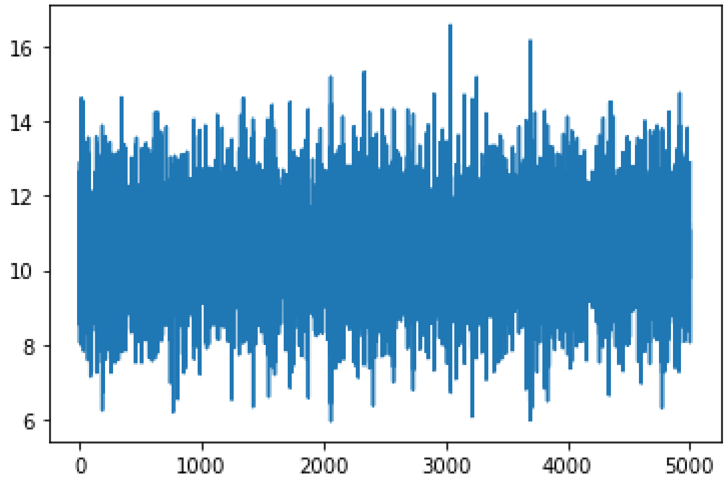
```
phis=[-0.7]
start_numbers=[1,2,3]
```

In [31]:

```
result=AR(p,c,phis)
plt.plot(pd.Series(result))
```

Out[31]:

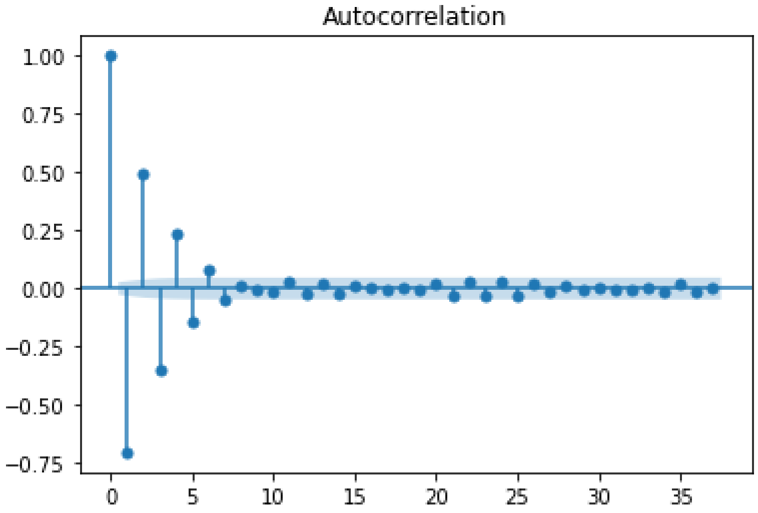
```
[<matplotlib.lines.Line2D at 0x20087753490>]
```

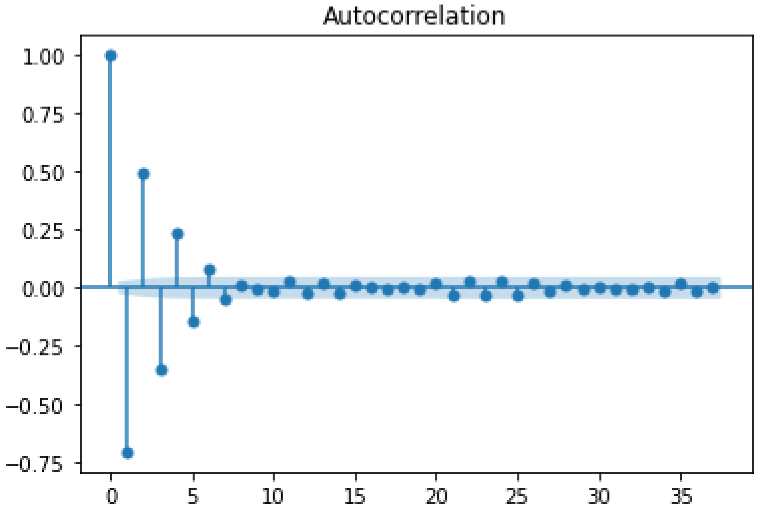


In [32]:

```
plot_acf(pd.Series(result))
```

Out[32]:

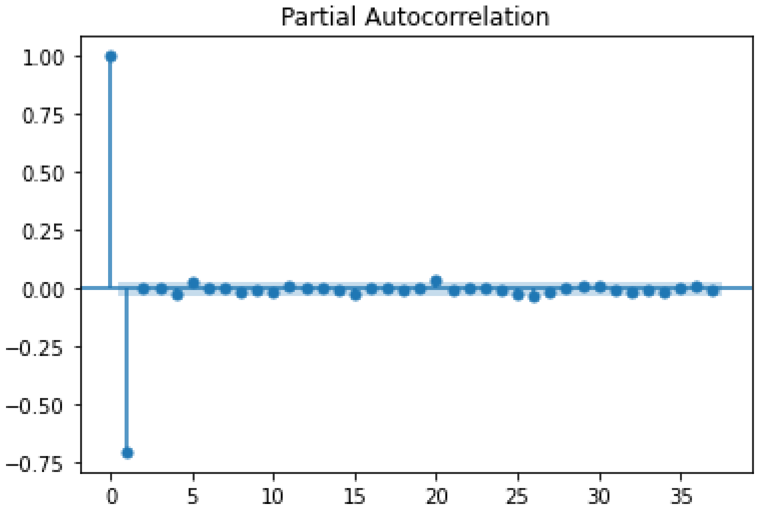


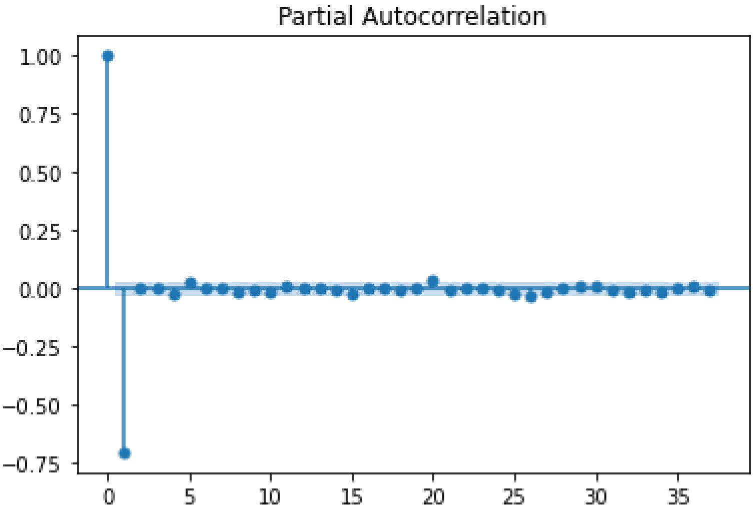


In [33]:

```
plot_pacf(pd.Series(result))
```

Out[33]:





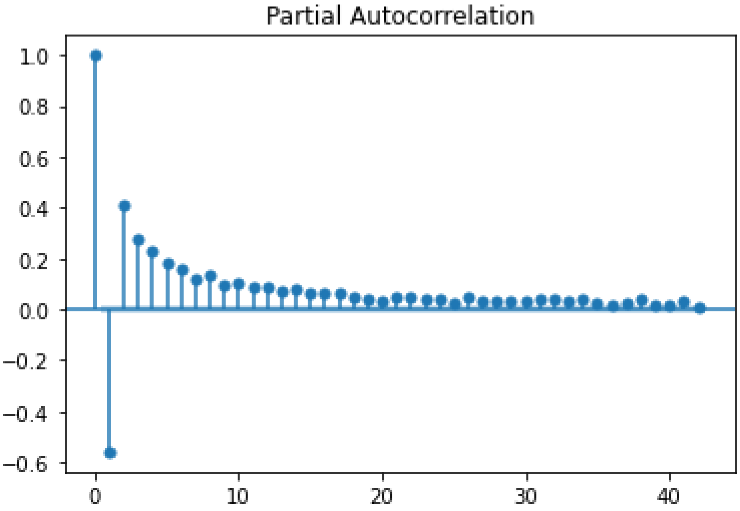
In [36]:

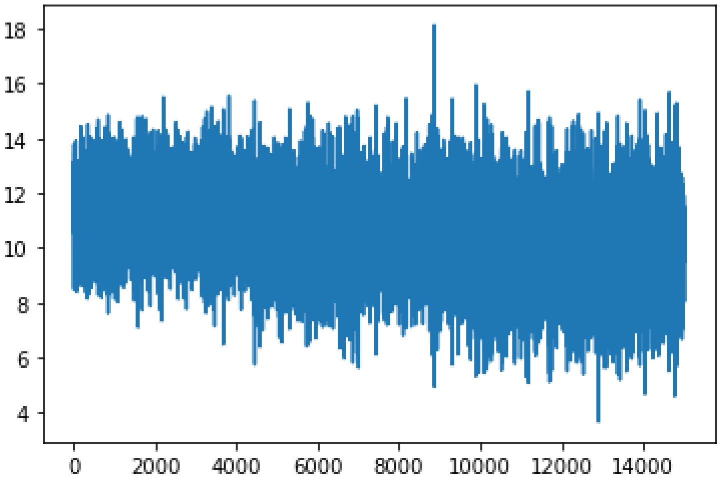
```
start_numbers=[1,2,3]
phis=[-0.8]
```

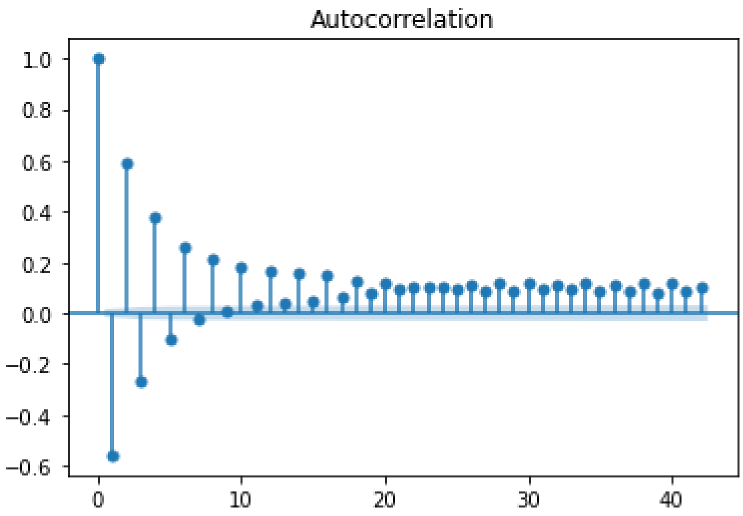
In [11]:

```
result=AR(p,c,phis)
plt.plot(pd.Series(result))
plot_acf(pd.Series(result))
plot_pacf(pd.Series(result))
```

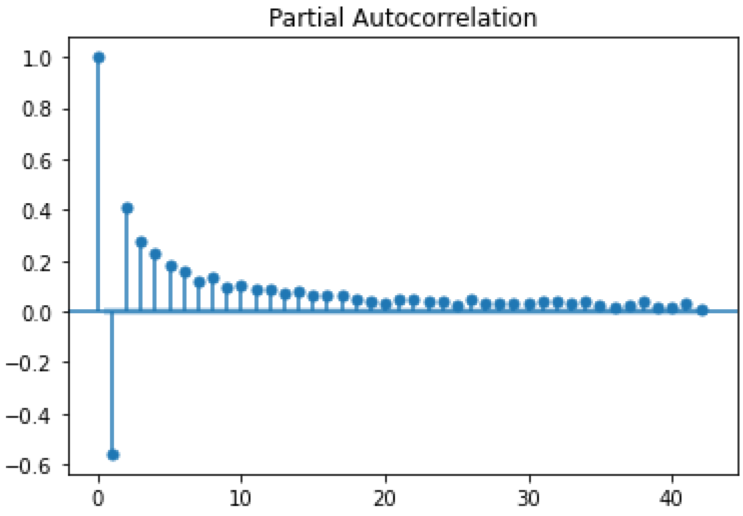
Out[11]:







Autocorrelation

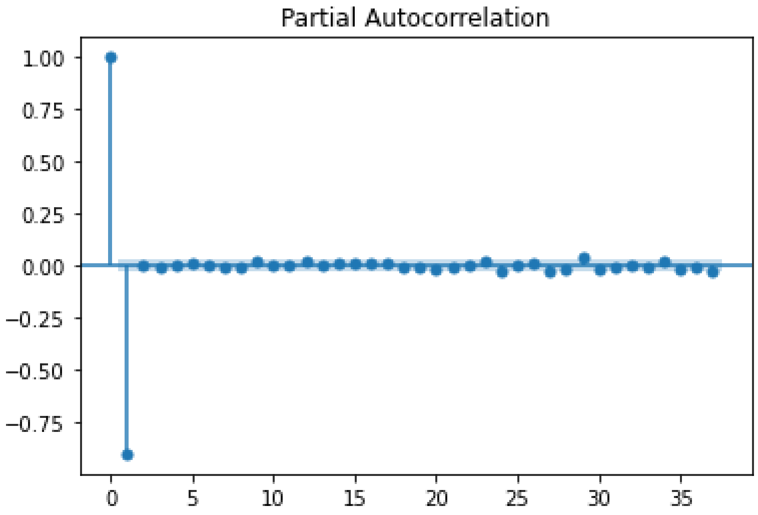


Partial Autocorrelation

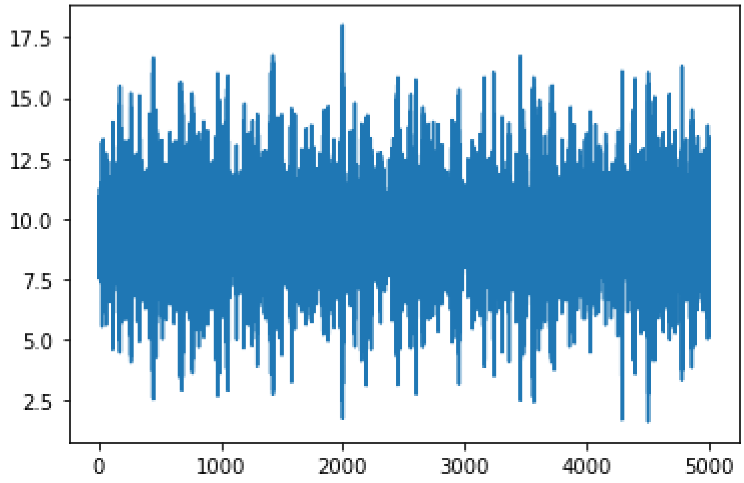
In [37]:

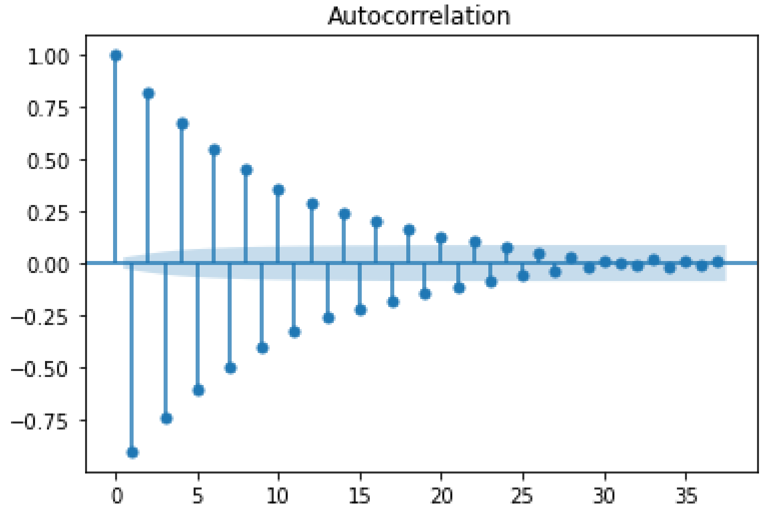
```
start_numbers=[1,2,3]
phis=[-0.9]
result=AR(p,c,phis)
plt.plot(pd.Series(result))
plot_acf(pd.Series(result))
plot_pacf(pd.Series(result))
```

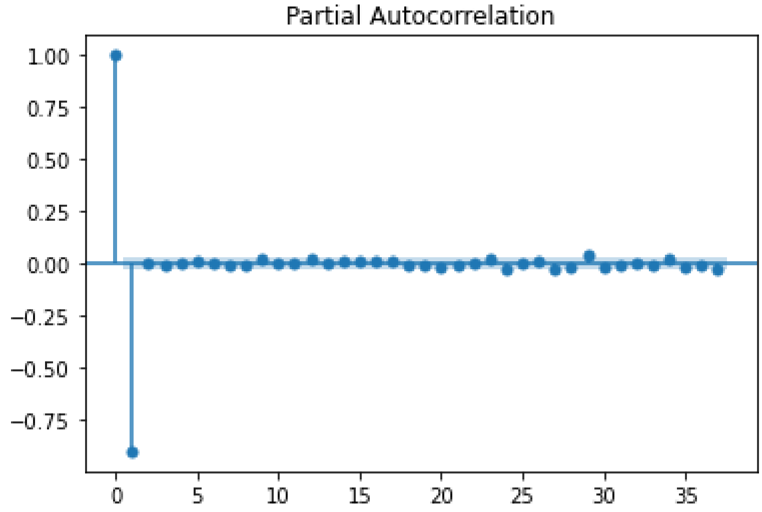
Out[37]:



Partial Autocorrelation







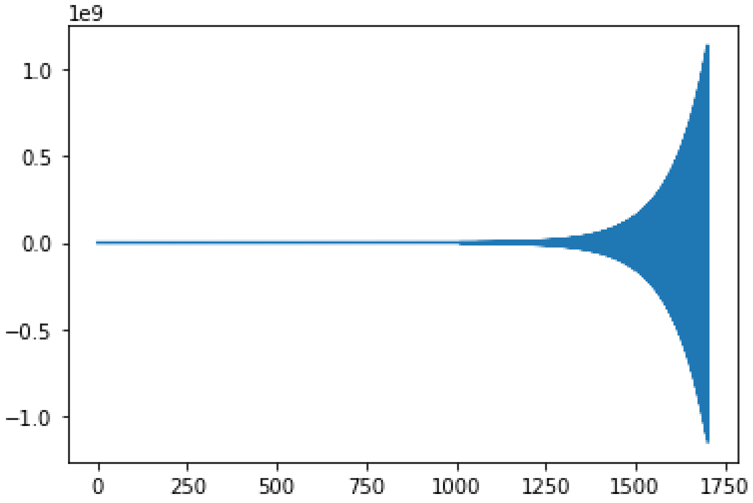
Are the generated time series stationary? What happens when |ϕ1| >

In [44]:

```
#With |phi|>1 time series becomes unstationary
start_numbers=[1,2,3]
phis=[-1.01]
simulation_length=2000
result=AR(p,c,phis)
plt.plot(pd.Series(result))
```

Out[44]:

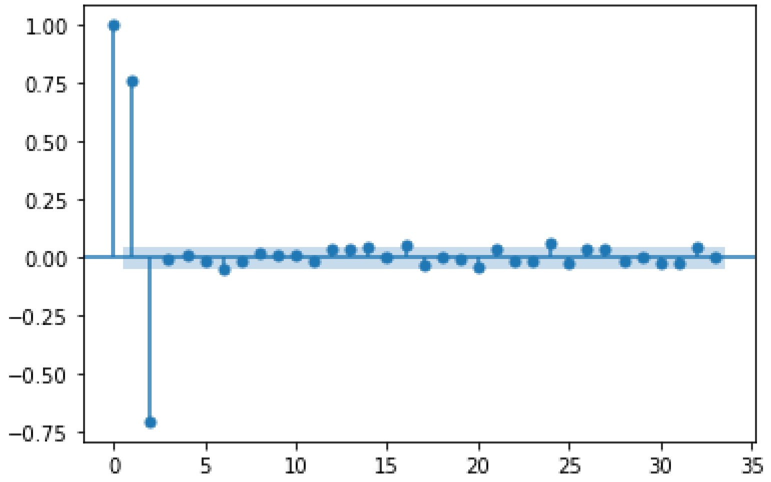
```
[<matplotlib.lines.Line2D at 0x2008daf6100>]
```

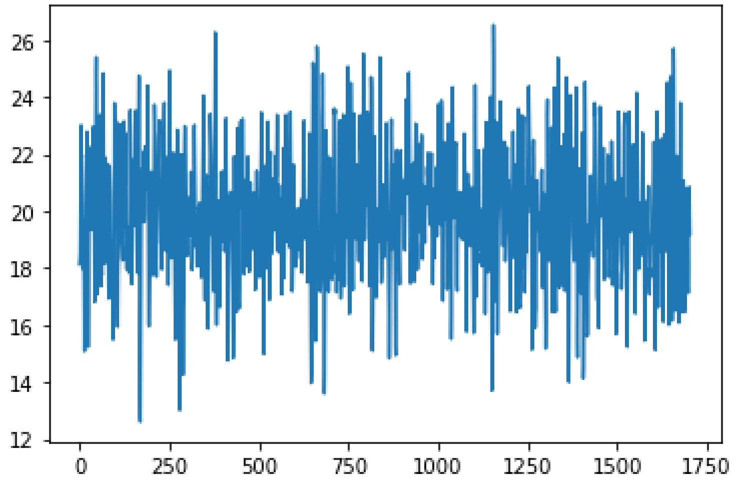


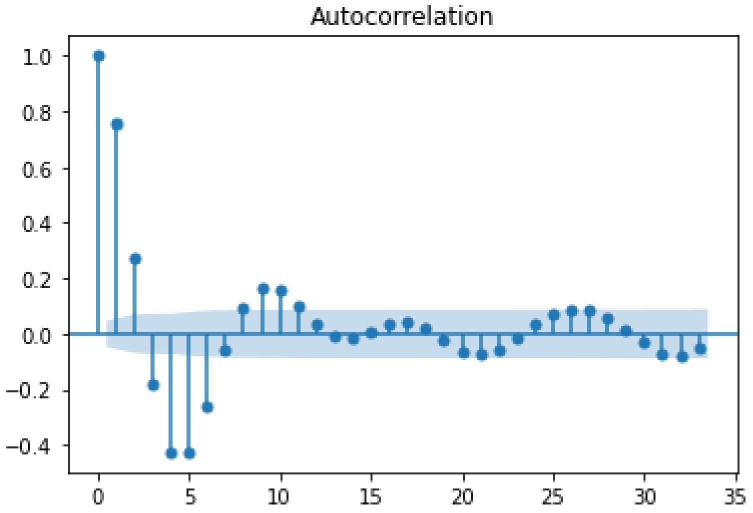
In [49]:

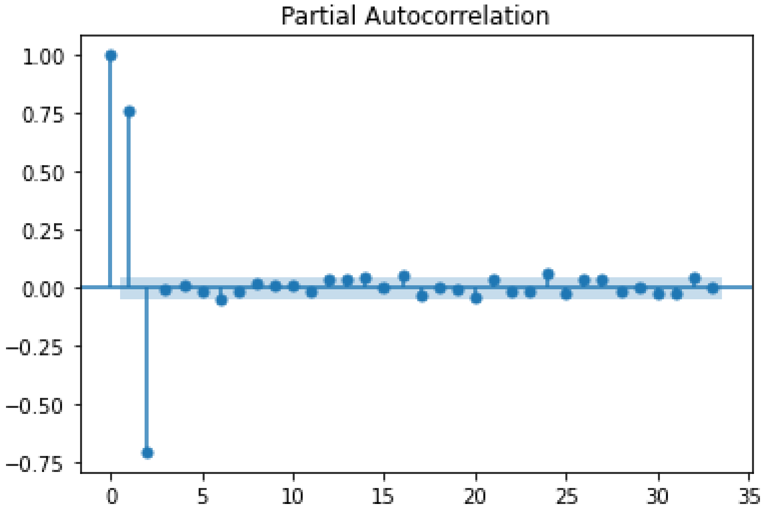
```
start_numbers=[1,2,3]
phis=[-0.7,1.3]
p=2
c=8
simulation_length=2000
result=AR(p,c,phis)
plt.plot(pd.Series(result))
plot_acf(pd.Series(result))
plot_pacf(pd.Series(result))
```

Out[49]:


Partial Autocorrelation







It's visible that for AR(2) Autocorellation is higher for shifts >1

```
In [ ]:
```