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

In [2]:
```
df=pd.read_excel('MeasurementData_Q1.xlsx')
```

In [3]:
```
df.head()
```

Out[3]:

	Year	Measurement
0	1984	539.9
1	1985	558.1
2	1986	620.1
3	1987	612.5
4	1988	640.6

In [4]:
```
df.index = np.arange(1, len(df) + 1)
df.reset_index(inplace=True)
```

In [5]:
```
df.head()
```

Out[5]:

	index	Year	Measurement
0	1	1984	539.9
1	2	1985	558.1
2	3	1986	620.1
3	4	1987	612.5
4	5	1988	640.6

In [7]:
```
df.shape
```

Out[7]:
```
(22, 3)
```

Q1 a)

In [8]:
```
time_period=11
```

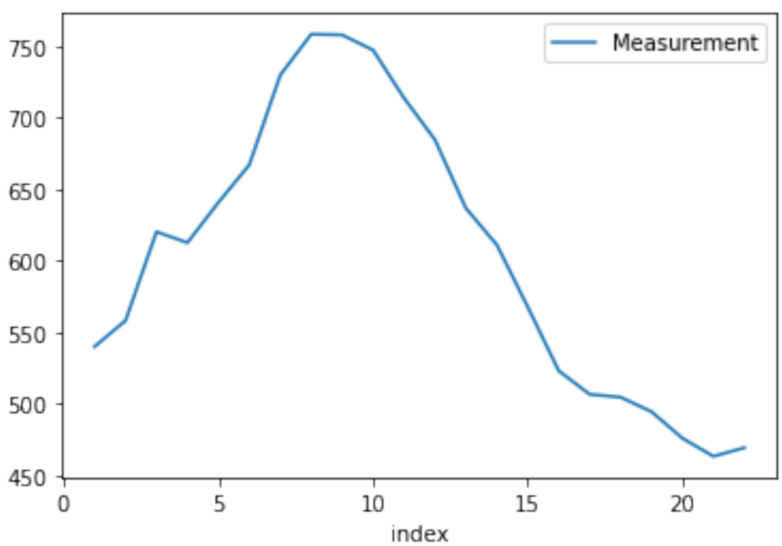
In [10]:
```
i=0
mov_avg=[]
while i < len(df.Measurement) - time_period + 1:
    moving_avg=np.sum(df.Measurement[i:i+time_period])/time_period
    mov_avg.append(moving_avg)
    i=i+1
```

In [14]:
```
print(mov_avg)
```

```
[667.6636363636363, 680.8090909090911, 687.9454545454547, 687.1181818181819,
683.0363636363637, 672.3454545454547, 657.7636363636365, 637.3000000000001,
613.3181818181819, 587.690909090909, 561.8818181818181, 539.6636363636363]
```

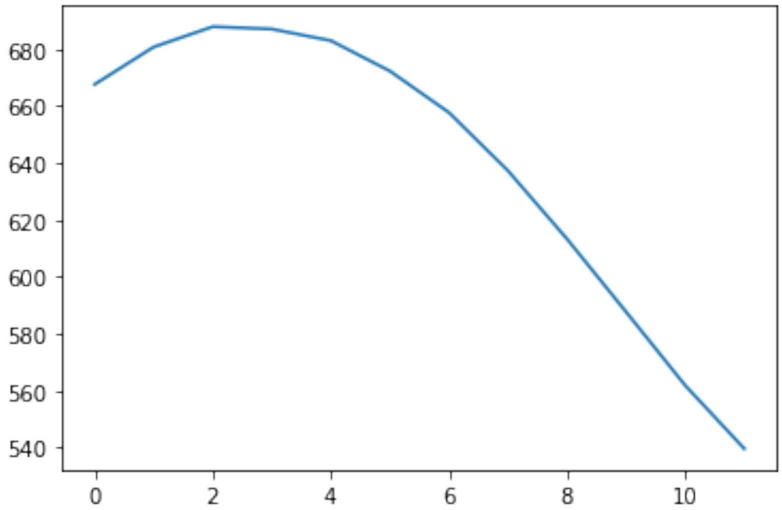
In [23]:
```
df.plot('index','Measurement')
```

Out[23]:
```
<AxesSubplot:xlabel='index'>
```



In [24]:
```
plt.plot(mov_avg)
```

Out[24]:
```
[<matplotlib.lines.Line2D at 0x7fb52d7ee0a0>]
```



Q1 b)

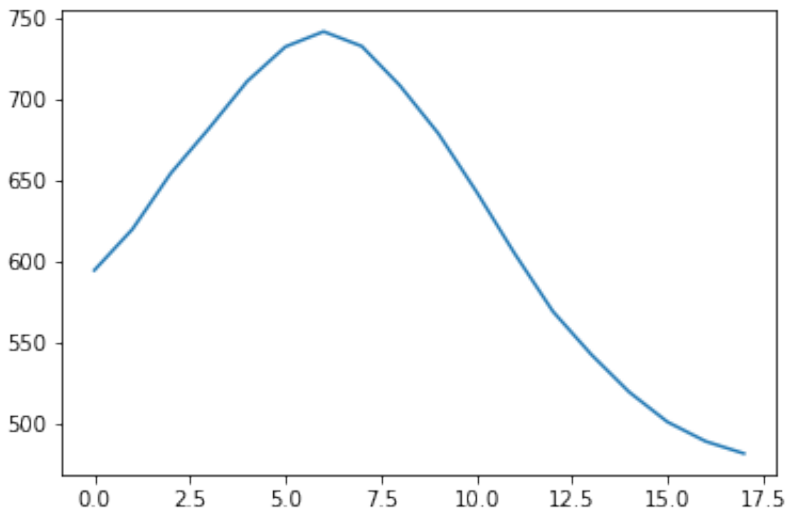
In [29]:
```
time_period2=5
i=0
mov_avg=[]
while i < len(df.Measurement) - time_period2 + 1:
    moving_avg=np.sum(df.Measurement[i:i+time_period2])/time_period2
    mov_avg.append(moving_avg)
    i=i+1
```

In [30]:
```
print(mov_avg)
```

```
[594.24, 619.6400000000001, 653.9399999999999, 681.5600000000001, 710.6, 731
.8999999999999, 741.24, 732.22, 707.9, 678.56, 642.66, 604.54, 568.939999999
9999, 542.52, 519.2, 500.84000000000003, 488.88, 481.41999999999996]
```

In [31]:
```
plt.plot(mov_avg)
```

Out[31]:
```
[<matplotlib.lines.Line2D at 0x7fb548f720a0>]
```



Q1 c)

After calculating the moving averages for two separate time periods 11 and 5. It can be observed that larger the time period more smoothed the data is but, a significant disadvantage is that we skip a lot of data points. On the other hand, when time period was 5 the graph looked somewhat similar to the original and few sharp points were smoothed out.

Q3

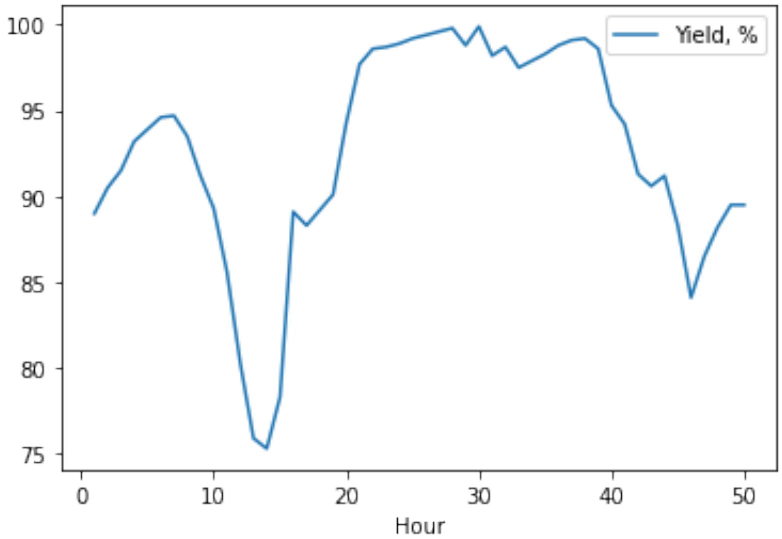
In [32]:
```
df2=pd.read_excel('Yield_Data.xlsx')
df2.head()
```

Out[32]:

	Hour	Yield, %
0	1	89.0
1	2	90.5
2	3	91.5
3	4	93.2
4	5	93.9

In [39]:
```
df2.plot('Hour','Yield, %')
```

Out[39]:
```
<AxesSubplot:xlabel='Hour'>
```



Q3 a)

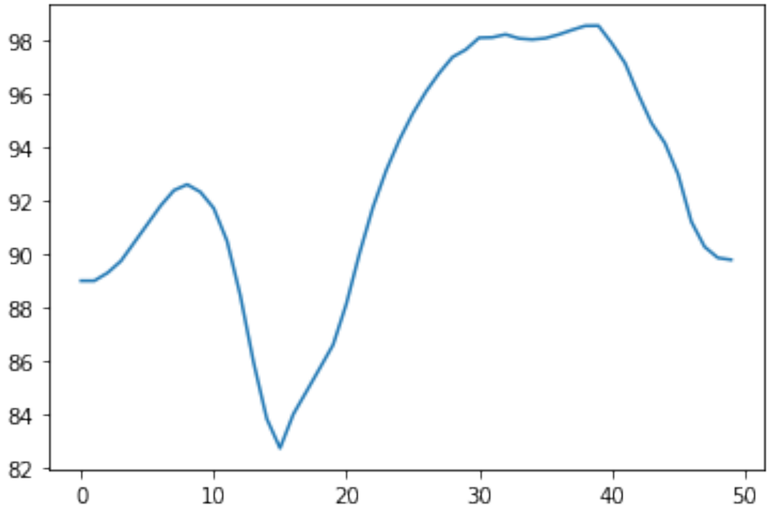
In [58]:
```
# print(Ft)
def calculateSmoothing(alpha, df2):
    Forecast=[]
    i=1
    Forecast.append(df2['Yield, %'][0]) #since first y0 is not given
    while i< len(df2['Yield, %']):
        Ft=alpha*df2['Yield, %'][i-1] + (1-alpha)*Forecast[i-1]
    #     df2['Yield, %'][i-1]
        #print(Ft)
        Forecast.append(Ft)
        i=i+1
    return Forecast

Forecast2=calculateSmoothing(0.2, df2)
print(Forecast2)
```

```
[89.0, 89.0, 89.30000000000001, 89.74000000000001, 90.43200000000002, 91.125
60000000002, 91.82048000000002, 92.39638400000001, 92.61710720000002, 92.333
68576000001, 91.72694860800001, 90.50155888640002, 88.46124710912002, 85.948
99768729603, 83.81919814983682, 82.71535851986945, 83.99228681589557, 84.853
82945271645, 85.72306356217317, 86.59845084973853, 88.13876067979082, 90.051
00854383267, 91.76080683506613, 93.1486454680529, 94.29891637444233, 95.2791
3309955387, 96.1033064796431, 96.80264518371449, 97.4021161469716, 97.681692
91757728, 98.12535433406184, 98.14028346724947, 98.25222677379958, 98.101781
41903967, 98.06142513523174, 98.1091401081854, 98.24731208654833, 98.4178496
6923868, 98.57427973539095, 98.57942378831277, 97.92353903065022, 97.1788312
2452018, 96.00306497961616, 94.92245198369294, 94.17796158695435, 93.0023692
6956349, 91.2218954156508, 90.27751633252065, 89.86201306601652, 89.78961045
281322]
```

In [57]:
```
plt.plot(Forecast2)
```

Out[57]: [<matplotlib.lines.Line2D at 0x7fb4f815b040>]



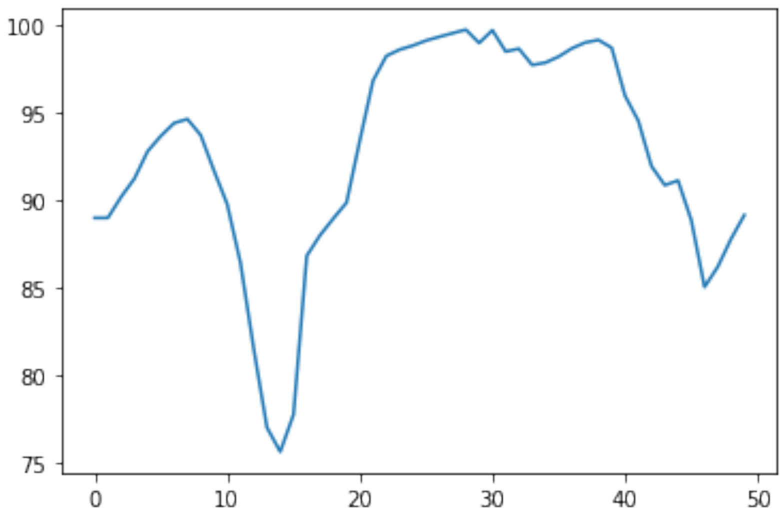
Q3 b)

In [59]:
```
Forecast3=calculateSmoothing(0.8, df2)
print(Forecast3)
```

```
[89.0, 89.0, 90.2, 91.24, 92.80799999999999, 93.6816, 94.41631999999998, 94.643264, 93.72865279999999, 91.70573056, 89.78114611199999, 86.4362292224, 81.52724584447999, 77.025449168896, 75.6450898337792, 77.76901796675584, 86.83380359335116, 88.00676071867022, 88.96135214373405, 89.8722704287468, 93.41445408574936, 96.84289081714988, 98.24857816342997, 98.609715632686, 98.8419431265372, 99.12838862530745, 99.34567772506149, 99.5491355450123, 99.74982710900247, 98.98996542180049, 99.7179930843601, 98.50359861687201, 98.6607197233744, 97.73214394467487, 97.86642878893498, 98.21328575778699, 98.68265715155741, 99.01653143031147, 99.16330628606231, 98.71266125721246, 95.98253225144248, 94.5565064502885, 91.9513012900577, 90.87026025801154, 91.13405205160231, 88.86681041032045, 85.05336208206408, 86.21067241641282, 87.80213448328256, 89.16042689665652]
```

In [60]:
```
plt.plot(Forecast3)
```

Out[60]: [<matplotlib.lines.Line2D at 0x7fb53d7ae730>]



Q3 c)

Below I have computed the Mean square difference for both values of lamda. It can be observed that lamda=0.2 produced a lower mean square difference than lamda=0.8 .

In [61]:
```
i=0
sum=0
MSE=0
while i< len(df2['Yield, %']):
    sum=((df2['Yield, %'][i] - Forecast2[i])**2)
    sum=sum+sum
    i=i+1
MSE=sum/len(df2['Yield, %'])
print(MSE)
```

```
0.0033549685751471632
```

In [62]:
```
i=0
sum=0
MSE=0
while i< len(df2['Yield, %']):
    sum=((df2['Yield, %'][i] - Forecast3[i])**2)
    sum=sum+sum
    i=i+1
MSE=sum/len(df2['Yield, %'])
print(MSE)
```

```
0.004612395700572945
```

In [ ]: