In [1]:
```
import numpy as np
import pylab as pl

data_992B=np.fromfile('sec1_992B', dtype=np.complex64)
```

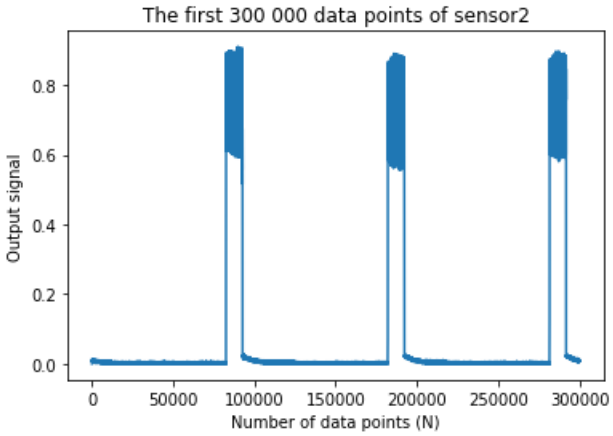
In [2]:
```
samples_int=np.abs(data_992B)
```

In [3]:
```
len(samples_int)
```

Out[3]:
```
26214400
```

In [11]:
```
pl.plot(samples_int[0:300000])
pl.xlabel('Number of data points (N)')
pl.ylabel('Output signal')
pl.title('The first 300 000 data points of sensor2')
```

Out[11]:
```
Text(0.5,1,'The first 300 000 data points of sensor2')
```



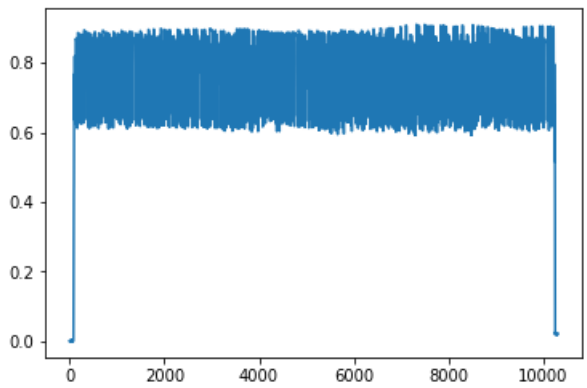
In [323]:
```
##m=data_992B[82400:92700]
m=data_992B[82400:92700]
```

In [324]:
```
len(m)
```

Out[324]:
```
10300
```

In [325]:
```
pl.plot(m)
```

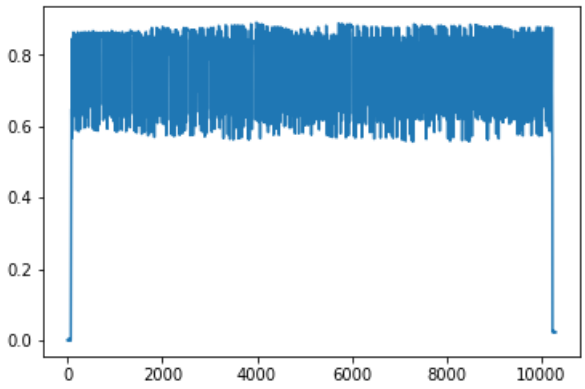
Out[325]:
```
[<matplotlib.lines.Line2D at 0x1575c35c0>]
```



In [326]:
```
m1=np.abs(data_992B[92700+89300:92700+89300+10300])
```

In [327]:
```
pl.plot(m1)
```

Out[327]:
```
[<matplotlib.lines.Line2D at 0x157668eb8>]
```



In [340]:
```
b=np.arange(1,3,1.)
```

In [341]:
```
type(b)
```

Out[341]:
```
numpy.ndarray
```

In [3]:
```
criteria=len(samples_int)
```

In [53]:
```
criteria
```

Out[53]:
```
26214400
```

In [ ]:
```
#number of datapoints=26214400/99600=263
```

In [ ]:
```
import matplotlib.pyplot as plt
import pylab as pl
import numpy as np
i=82400
n=0
mean1=np.arange(1,264,1.)
while i < criteria:
    mean1=samples_int[i:i+10300]
    plt.figure()
    plt.plot(mean1)
    #plt.savefig('plot' + str(n) + '.png')
    i=i+99600
    n=n+1
```

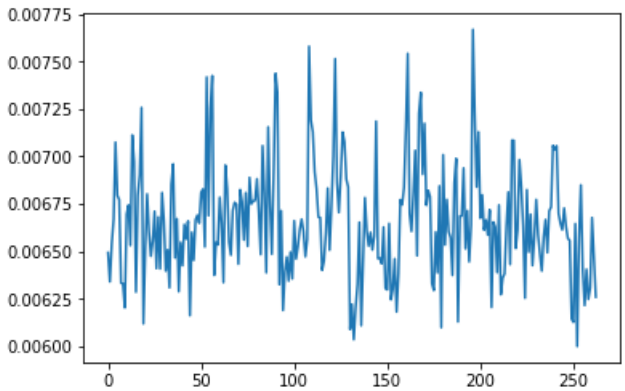
In [23]:
```
import matplotlib.pyplot as plt
d=0
i=0
mean1=np.arange(264.)
std1=np.arange(264.)
var1=np.arange(264.)
while i<=criteria:

###################################
    if samples_int[i]>=0.05:
        mean1[d]=np.mean(samples_int[i:i+10172])
        std1[d]=np.std(samples_int[i:i+10172])
        var1[d]=np.var(samples_int[i:i+10172])
        d=d+1
        i=i+10800
        #plt.figure()
        #plt.plot(mean1)
        #plt.savefig('plot' + str(n) + '.png')
###################################
    i=i+1
```

```
---------------------------------------------------------------------------
IndexError                                Traceback (most recent call last)
<ipython-input-23-09a4e785f037> in <module>()
      8
      9 ###################################
---> 10     if samples_int[i]>=0.05:
     11         mean1[d]=np.mean(samples_int[i:i+10172])
     12         std1[d]=np.std(samples_int[i:i+10172])

IndexError: index 26214400 is out of bounds for axis 0 with size 26214400
```

In [25]:
```
mean1=mean1[:-1]
std1=std1[:-1]
var1=var1[:-1]
```

In [28]:
```
pl.plot(var1)
```

Out[28]:
```
[<matplotlib.lines.Line2D at 0x12cf9e438>]
```



In [46]:
```
dataset={'Mean1':mean1, 'Standard_Deviation1':std1, 'Variance1':var1}
```

In [47]:
```
import pandas as pd
df1=pd.DataFrame(data=dataset)
```

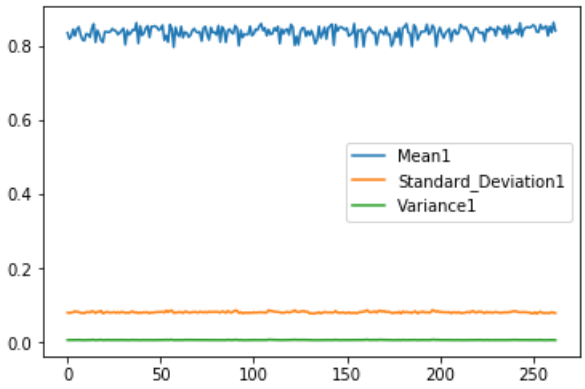
In [48]:
```
df1.head()
```

Out[48]:

	Mean1	Standard_Deviation1	Variance1
0	0.835184	0.080572	0.006492
1	0.819339	0.079616	0.006339
2	0.825923	0.080863	0.006539
3	0.845044	0.081661	0.006668
4	0.827356	0.084111	0.007075

In [50]:
```
df1.plot()
```

Out[50]:
```
<matplotlib.axes._subplots.AxesSubplot at 0x129cdb470>
```



In [51]:
```
import numpy as np
import pylab as pl

data_992B=np.fromfile('sec1_9952', dtype=np.complex64)
```

In [52]:
```
samples_int=np.abs(data_992B)
criteria=len(samples_int)
```

In [39]:
```
import matplotlib.pyplot as plt
d=0
i=0
mean2=np.arange(264.)
std2=np.arange(264.)
var2=np.arange(264.)
while i<=criteria:

##################################
    if samples_int[i]>=0.05:
        mean2[d]=np.mean(samples_int[i:i+10172])
        std2[d]=np.std(samples_int[i:i+10172])
        var2[d]=np.var(samples_int[i:i+10172])
        d=d+1
        i=i+10800
        #plt.figure()
        #plt.plot(mean1)
        #plt.savefig('plot' + str(n) + '.png')
##################################
    i=i+1
```

```
---------------------------------------------------------------------------
IndexError                                Traceback (most recent call last)
<ipython-input-39-1af73ba05e37> in <module>()
      8
      9 ##################################
---> 10     if samples_int[i]>=0.05:
     11         mean2[d]=np.mean(samples_int[i:i+10172])
     12         std2[d]=np.std(samples_int[i:i+10172])

IndexError: index 26214400 is out of bounds for axis 0 with size 26214400
```

In [40]:
```
mean2=mean2[:-1]
std2=std2[:-1]
var2=var2[:-1]
```

In [53]:
```
dataset={'Mean2':mean2, 'Standard_Deviation2':std2, 'Variance2':var2}

import pandas as pd
df2=pd.DataFrame(data=dataset)

df2.head()
```

Out[53]:

	Mean2	Standard_Deviation2	Variance2
0	0.850708	0.073740	0.005438
1	0.845545	0.075150	0.005647
2	0.843262	0.071994	0.005183
3	0.847141	0.076289	0.005820
4	0.843402	0.074116	0.005493

In [54]:
```
df2.head()
```

Out[54]:

	Mean2	Standard_Deviation2	Variance2
0	0.850708	0.073740	0.005438
1	0.845545	0.075150	0.005647
2	0.843262	0.071994	0.005183
3	0.847141	0.076289	0.005820
4	0.843402	0.074116	0.005493

In [55]:
```
data_add=pd.concat([df1,df2],axis=1)
```

In [65]:
```
data_add.head()
```

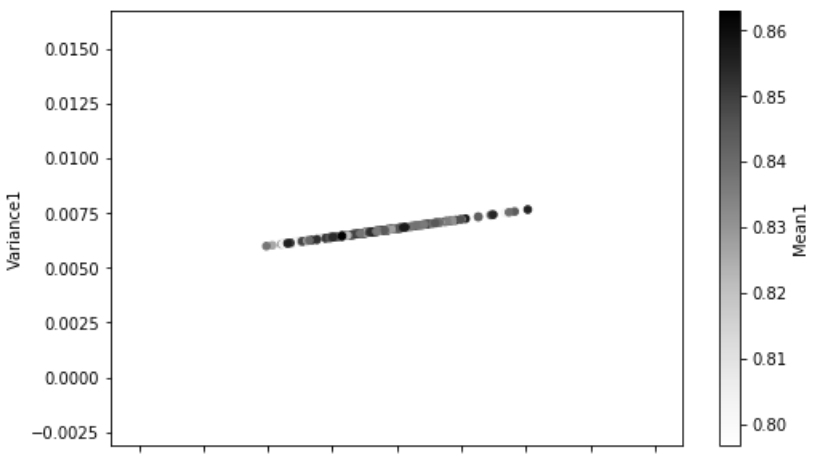
Out[65]:

	Mean1	Standard_Deviation1	Variance1	Mean2	Standard_Deviation2	Variance2
0	0.835184	0.080572	0.006492	0.850708	0.073740	0.005438
1	0.819339	0.079616	0.006339	0.845545	0.075150	0.005647
2	0.825923	0.080863	0.006539	0.843262	0.071994	0.005183
3	0.845044	0.081661	0.006668	0.847141	0.076289	0.005820
4	0.827356	0.084111	0.007075	0.843402	0.074116	0.005493

In [66]:
```
data_add.to_csv('new_02_2019.csv')
```

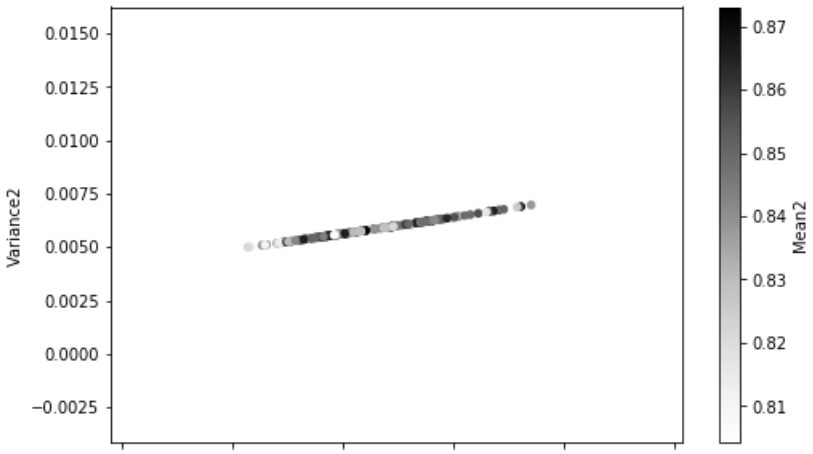
In [63]:
```
data_add.plot.scatter(x='Standard_Deviation1',y='Variance1', c='Mean1', figsize=(8,5))
```

Out[63]:
```
<matplotlib.axes._subplots.AxesSubplot at 0x12aa2c320>
```



In [64]:
```
data_add.plot.scatter(x='Standard_Deviation2',y='Variance2', c='Mean2', figsize=(8,5))
```

Out[64]:
```
<matplotlib.axes._subplots.AxesSubplot at 0x12933d4e0>
```



In [68]:
```
%pwd
```

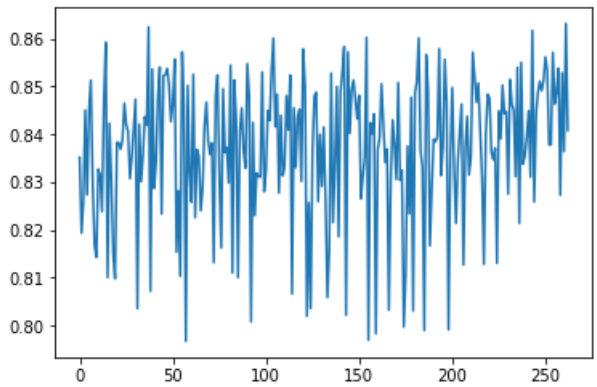
Out[68]:
```
'/Users/zulfidinkhodzhaev/Documents/1.PYTHON/SPECTROGRAM/Reading Sensor'
```

In [ ]:
```
```

In [20]:
```
mean1_new=mean1[:-1]
```

In [22]:
```
pl.plot(mean1_new)
```

Out[22]:
```
[<matplotlib.lines.Line2D at 0x12a750780>]
```



In [ ]:
```
import matplotlib.pyplot as plt
d=0
i=0
mean1=np.arange(264.)
while i<=criteria:

##################################
    if samples_int[i]>=0.05:
        plt.figure()
        plt.plot(samples_int[i:i+10172])

        i=i+10800

        #plt.savefig('plot' + str(n) + '.png')
##################################
    i=i+1
```

In [373]:
```
test=np.arange(1,3,1.)
```

In [374]:
```
test=cle
```

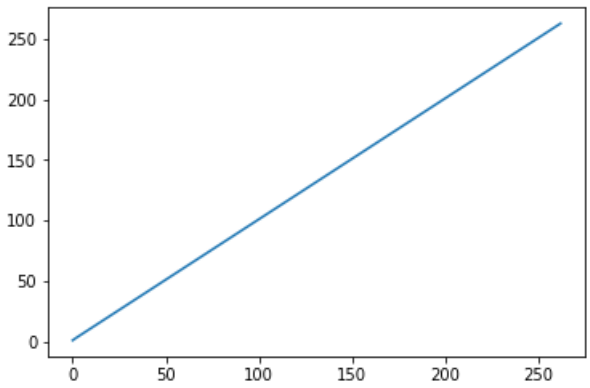
```
---------------------------------------------------------------------------
NameError                                 Traceback (most recent call last)
<ipython-input-374-c0e9b357634d> in <module>()
----> 1 test=clear

NameError: name 'clear' is not defined
```

In [372]:
```
type(test)
```

Out[372]:
```
NoneType
```

In [365]:
```
pl.plot(mean1)
```

Out[365]:
```
[<matplotlib.lines.Line2D at 0x16aaff160>]
```



In [ ]:
```
m=0
for i in m3:
    print(m3[m])
    m=m+1
    print(m)
```

In [142]:
```
len(m3)
```

Out[142]:
```
10260
```

In [29]:
```
d=0
c=0
g=0
i=0
criteria=len(samples_int)
mean1=np.arange(264.)

for i in samples_int:
    while samples_int[d]>=0.05:
            data[c]=samples_int[d]
            d=d+1
            c=c+1
        final[g]=np.mean(data)
        plt.figure()
        plt.plot(mean1)
        g=g+1
    d=d+1
```

```
---------------------------------------------------------------------------
NameError                                 Traceback (most recent call last)
<ipython-input-29-c972d5e18ca9> in <module>()
      9     while samples_int[d]>=0.05:
     10         while samples_int[d]>=0.05:
---> 11             data[c]=samples_int[d]
     12             d=d+1
     13             c=c+1

NameError: name 'data' is not defined
```

In [ ]:
```
for i in samples_int:
    while samples_int[d]>=0.05:
        print(d)
        d=d+1
        break
```

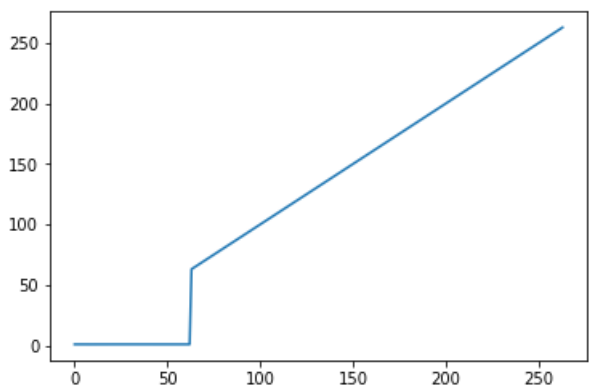
In [ ]:
```
d=0
i=0
mean1=np.arange(264.)
while i<=criteria:
##################################
    if samples_int[i]>=0.05:
        mean1[d]=samples_int[i]
        np.mean(samples_int[i])
        d=d+1

##################################
    i=i+1
```

In [7]:
```
criteria
```

Out[7]:
```
26214400
```

In [6]:
```
pl.plot(mean1)
```

Out[6]:
```
[<matplotlib.lines.Line2D at 0x131b089e8>]
```



In [ ]:
```
mean1
```

In [ ]:
```
d=0
i=0
mean1=np.arange(264.)
while i<=criteria:
    print(i)
##################################
    if i==3:
        i=i+4

##################################
    i=i+1
```

In [62]:
```
mean1
```

Out[62]:
```
array([0.83554846, 0.81969142, 0.01350524, 0.82629645, 0.84542286,
       0.82772511, 0.84525484, 0.85169631, 0.83028567, 0.01478213,
       0.81711966, 0.01301695, 0.81466091, 0.83309376, 0.83189112,
       0.82419813, 0.84680283, 0.01456248, 0.85959244, 0.01720753,
       0.81040645, 0.0138886 , 0.84268206, 0.83094633, 0.81536889,
       0.8101542 , 0.01114801, 0.83864427, 0.01118032, 0.83859688,
       0.0145643 , 0.83728772, 0.83922392, 0.8468371 , 0.01276129,
       0.84243381, 0.84104168, 0.83112502, 0.83565933, 0.01262778,
       0.84142852, 0.01387932, 0.84767121, 0.80393821, 0.01217223,
       0.84244984, 0.83051652, 0.83607584, 0.84405893, 0.84227425,
       0.86278582, 0.01395936, 0.80756044, 0.85402679, 0.82902539,
       0.01154254, 0.83463579, 0.84818345, 0.01275108, 0.85440564,
       0.01415585, 0.82375979, 0.85260046, 0.8528322 , 0.01611489,
       0.85414594, 0.01025714, 0.85046762, 0.01386897, 0.8430301 ,
       0.84678513, 0.85608548, 0.01536437, 0.81575167, 0.82859278,
       0.81072247, 0.85759437, 0.84639525, 0.79710627, 0.85058653,
       0.8325578 , 0.01487915, 0.82624382, 0.85288054, 0.82296145,
       0.01149822, 0.83719575, 0.83578354, 0.82441372, 0.8290807 ,
       0.84203768, 0.84710062, 0.83904451, 0.83615476, 0.838516  ,
       0.01322465, 0.81350982, 0.01311385, 0.84854215, 0.85278189,
       0.83137482, 0.81663042, 0.84986317, 0.01322281, 0.83653677,
       0.01358314, 0.83761984, 0.83024198, 0.85484439, 0.81141007,
       0.85167414, 0.01413264, 0.84341317, 0.01421092, 0.81045181,
       0.84109491, 0.84585309, 0.83686316, 0.01575153, 0.8333143 ,
       0.85514575, 0.8483628 , 0.80117249, 0.84282303, 0.82339394,
       0.83230579, 0.83150506, 0.8314538 , 0.01215266, 0.85337883,
       0.0128744 , 0.82834715, 0.01367204, 0.83268481, 0.01202409,
       0.84531671, 0.84328973, 0.85362399, 0.86048615, 0.84193635,
       0.01178162, 0.84873652, 0.01354109, 0.82812262, 0.84435731,
       0.8316943 , 0.83348584, 0.84844506, 0.84130603, 0.85273701,
       0.80698413, 0.84586835, 0.83337158, 0.01436828, 0.84403574,
       0.84559393, 0.83054888, 0.85823143, 0.85020411, 0.01209114,
       0.80235976, 0.82603896, 0.80400574, 0.83739066, 0.84836042,
       0.01216118, 0.84915525, 0.82632202, 0.84032458, 0.01204561,
       0.82950574, 0.84187436, 0.82670093, 0.80624557, 0.01132019,
       0.81622118, 0.01095339, 0.8531543 , 0.82197112, 0.83412862,
       0.85036492, 0.81896102, 0.01546995, 0.84912372, 0.01231319,
       0.85338193, 0.85868561, 0.80255312, 0.85757172, 0.01329   ,
       0.8405655 , 0.01314643, 0.84973657, 0.85172254, 0.84765941,
       0.84357423, 0.01412004, 0.84850854, 0.01433972, 0.82692009,
       0.83150363, 0.83562458, 0.86060852, 0.01137681, 0.79733682,
       0.01070559, 0.84276211, 0.01184075, 0.84048176, 0.84462273,
       0.79870301, 0.83713317, 0.83974481, 0.85094422, 0.84247983,
       0.83449155, 0.83731019, 0.8035866 , 0.01163842, 0.83099216,
       0.84338999, 0.83814245, 0.0160657 , 0.83095151, 0.85118741,
       0.83076286, 0.01409797, 0.83284903, 0.01536298, 0.80011904,
       0.81125861, 0.01009267, 0.83792675, 0.82379729, 0.84812278,
       0.80342555, 0.01214668, 0.84897679, 0.01405255, 0.85152853,
       0.01391432, 0.86050081, 0.83701503, 0.8334468 , 0.79938346,
       0.85698235, 0.01519783, 0.84967035, 0.81701589, 0.01128049,
       0.82902962, 0.83932787, 0.8388477 , 0.83986712, 0.01383014,
       0.85828096, 0.83174461, 0.83789176, 0.8560586 , 0.84691942,
       0.79953748, 0.83569843, 0.01203884, 0.85014206])
```

In [40]:
```
d=0
while samples_int[d]>=0.05:
    print(samples_int[d])
    d=d+1
```

In [48]:
```
d
```

Out[48]:
```
0
```

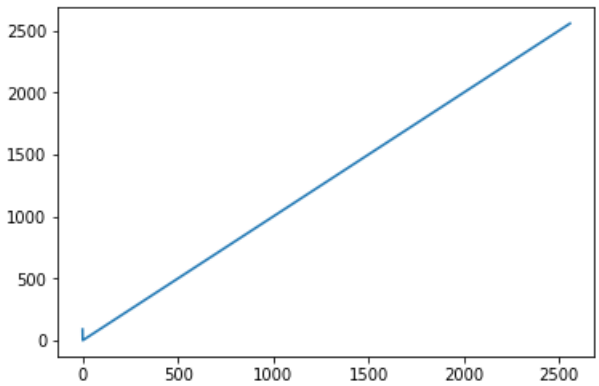
In [18]:
```
final=np.arange(5.0)
```

In [21]:
```
final
```

Out[21]:
```
array([ 0. ,  1. ,  2. , 55.5,  4. ])
```

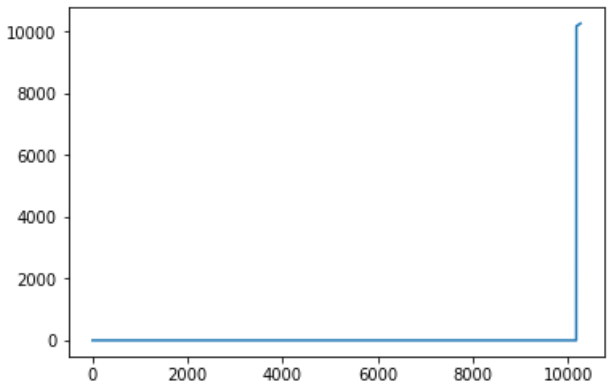
In [258]:
```
i=0
n=0
mean1=np.array(np.arange(1,2560,1.))
while i < criteria:
    mean1[n]=np.mean(samples_int[i:i+2000000])
    i=i+2000000
    n=n+1
```

Out[258]:
```
[<matplotlib.lines.Line2D at 0x14dac8e80>]
```



In [228]:
```
pl.plot(data)
```

Out[228]:
```
[<matplotlib.lines.Line2D at 0x141132828>]
```



In [234]:
```
d
```

Out[234]:
```
{'group0': 4, 'group1': 4, 'group2': 4}
```

In [179]:
```
m=np.arange(0,len(m3))
```

In [182]:
```
type(m)
```

Out[182]:
```
numpy.ndarray
```

In [189]:
```
m3[10259]
```

Out[189]:
```
0.023244781
```

In [ ]:
```
d=0
for i in m3:
    print(m3[d])
    d=d+1
```

In [ ]:
```

```