In [22]:

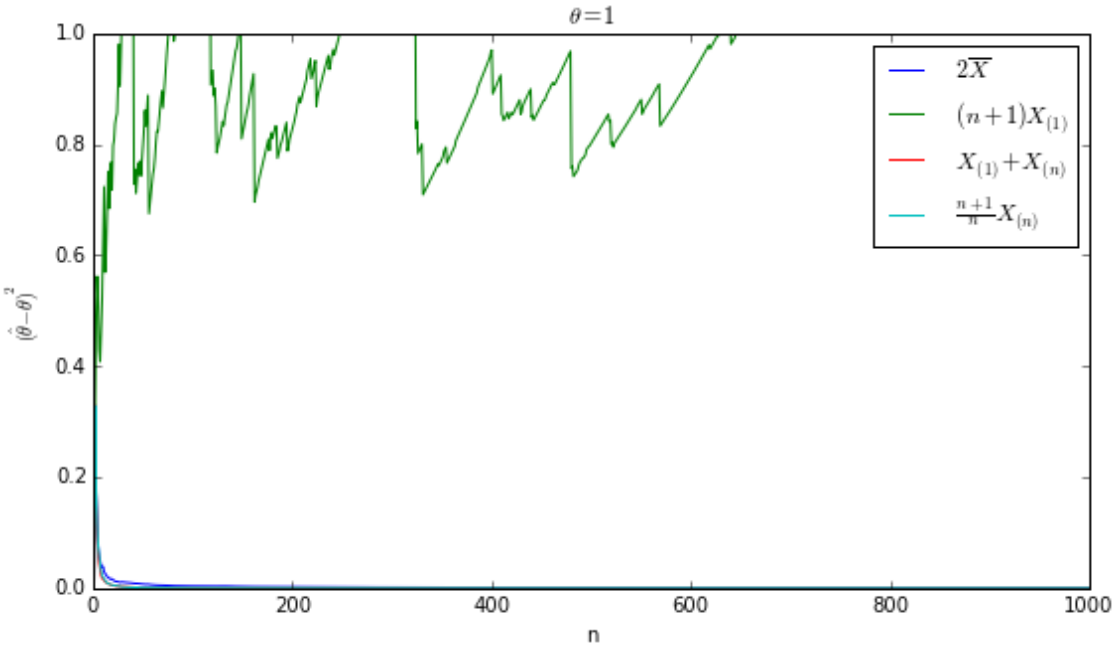
```
import pandas as pd
import numpy as np
import matplotlib.pyplot as plt
import scipy.stats as sps
import math
%matplotlib inline
```

In [23]:

```
N = 1000
M = 100

# theta - параметр мета
# scale - размер
def showGraphs(theta, scale) :
    x = np.arange(1, N + 1)
    y1 = np.zeros(N) # 2X
    y2 = np.zeros(N) # (n + 1) * X(1)
    y3 = np.zeros(N) # X(1) + X(n)
    y4 = np.zeros(N) # ((n + 1)/n) * X(n)

    for i in range(0, M):
        sample = sps.uniform(0, theta).rvs(size=N)
        for n in range(1, N) :
            part = sample[:n]
            y1[n] += (theta \
                      - (part.mean() * 2.0)) ** 2.0
            y2[n] += (theta \
                      - ((n + 1) * part.min())) ** 2.0
            y3[n] += (theta \
                      - (part.min() + part.max())) ** 2.0
            y4[n] += (theta \
                      - ((n + 1) / n * part.max())) ** 2.0

    plt.figure(figsize=(9, 5))
    plt.plot(x, y1 / M, label=' $2\overline{X}$')
    plt.plot(x, y2 / M, label=' $(n + 1)X_{(1)}$')
    plt.plot(x, y3 / M, label=' $X_{(1)} + X_{(n)}$')
    plt.plot(x, y4 / M, label=' $\\frac{n + 1}{n}X_{(n)}$')
    plt.legend()
    plt.ylim((0, scale))
    plt.xlabel('n')
    plt.ylabel('${(\\hat{\\theta}-\\theta)}^{2}$')
    plt.title('$\\theta=' + str(theta) + '$' )
    plt.show()
```

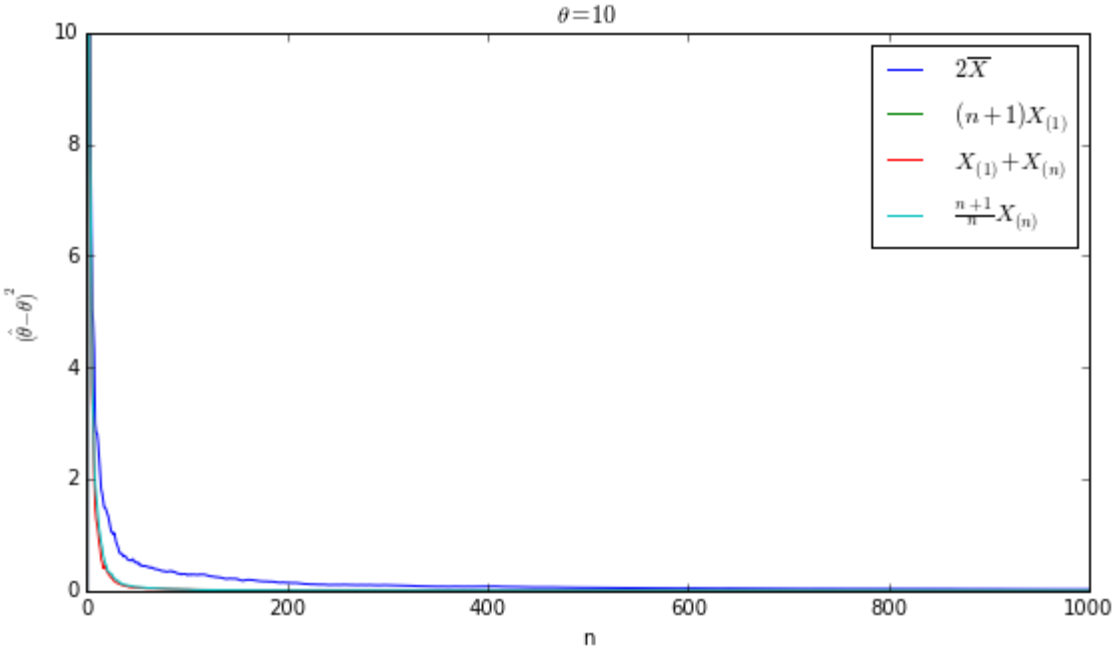
In [24]:

```
showGraphs(1, 1)
```



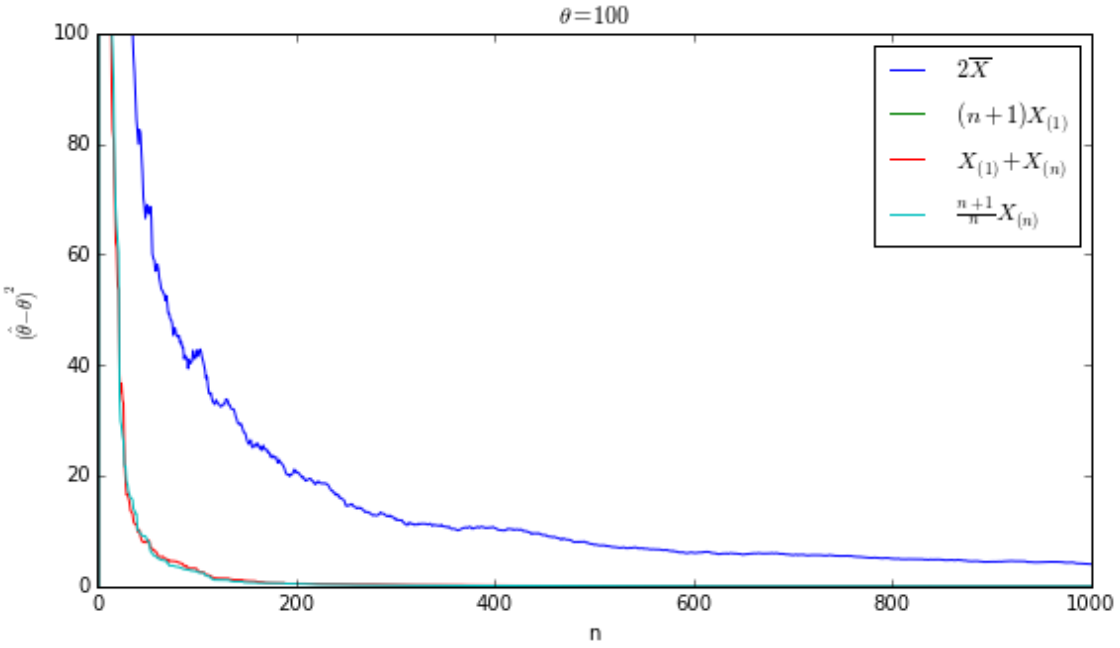
In [25]:

```
showGraphs(10, 10)
```



In [26]:

```
showGraphs(100, 100)
```



Ну, тут в общем то подтверждаются выводы из 1го номера 1го задания

In [ ]:

```

```