In [152]:

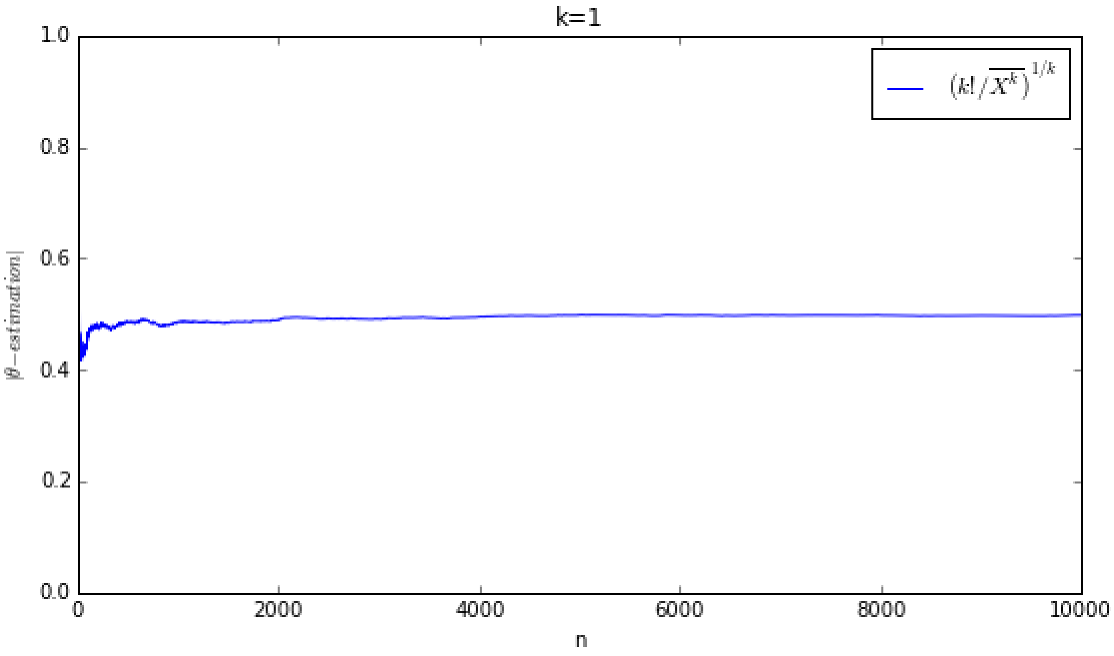
```
import pandas as pd
import numpy as np
import matplotlib.pyplot as plt
import scipy.stats as sts
import math
%matplotlib inline
```

In [180]:

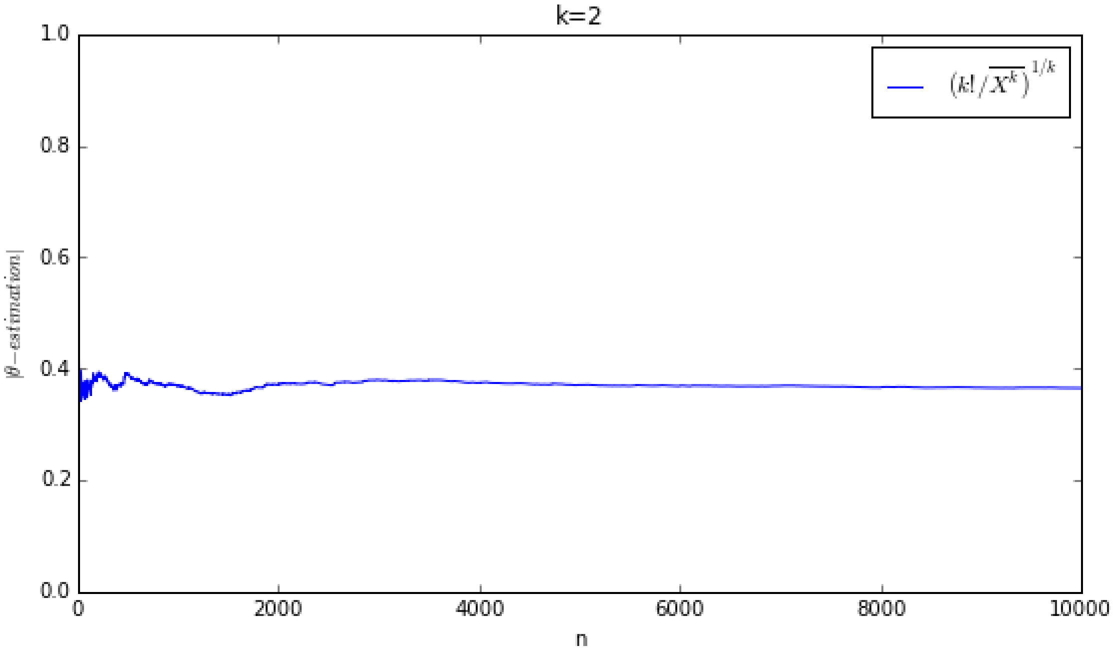
```
N = 10000
theta = 1.0

# k - константа из оценки
# scale - размер
def showGraphs(k, scale):
    global theta
    sample = sts.expon(theta).rvs(N)

    x = np.arange(1, N + 1)
    y = np.zeros(N)

    for n in range(1, N):
        samplePart = sample[:n]
        y[n] = abs(theta - (float(math.factorial(k)) \
                            / (samplePart ** k).mean()) ** (1.0 / k))

    plt.figure(figsize=(9, 5))
    plt.plot(x, y, label='${(k!/\overline{X^{k}})}^{1/k}$')
    plt.legend()
    plt.ylim((0, scale))
    plt.xlabel('n')
    plt.ylabel('$|\\theta - estimation|$')
    plt.title('k=' + str(k))
    plt.show()
    print "Delta:", y[-1]
```

In [174]:

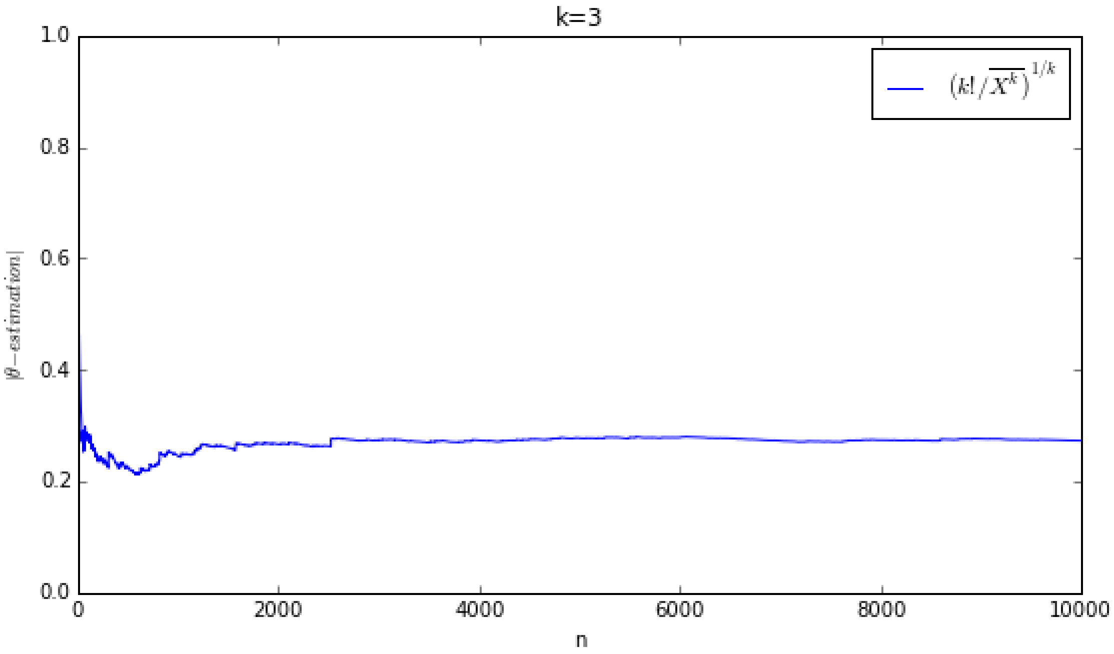
```
for k in range(1, 6):
    showGraphs(k, 1)
```



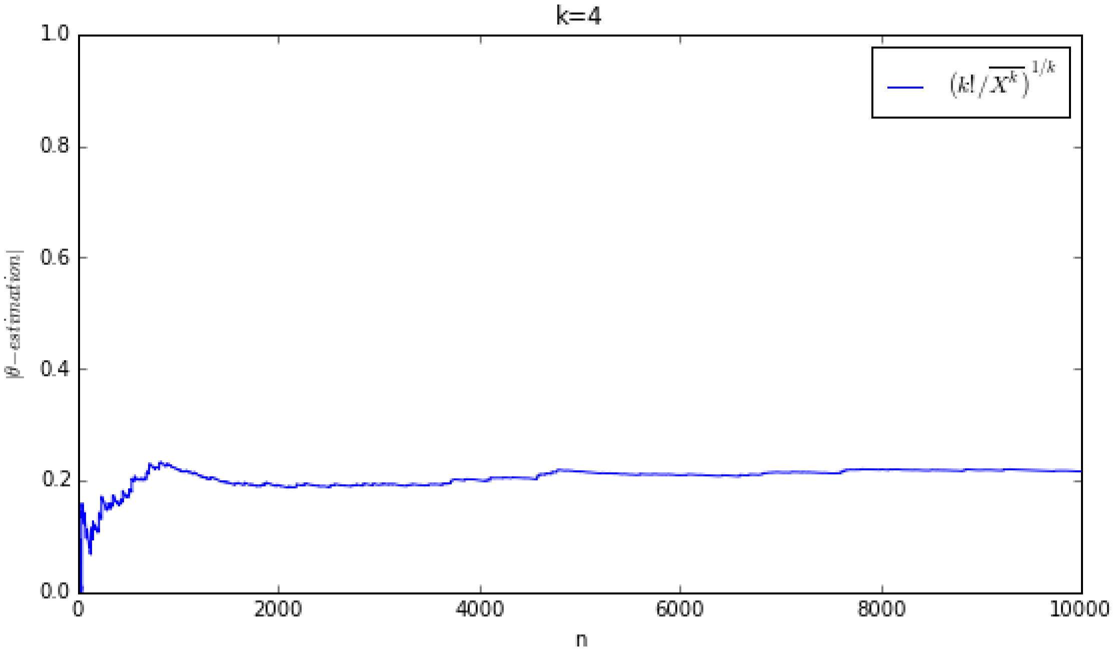
```
Delta: 0.498324545865
```



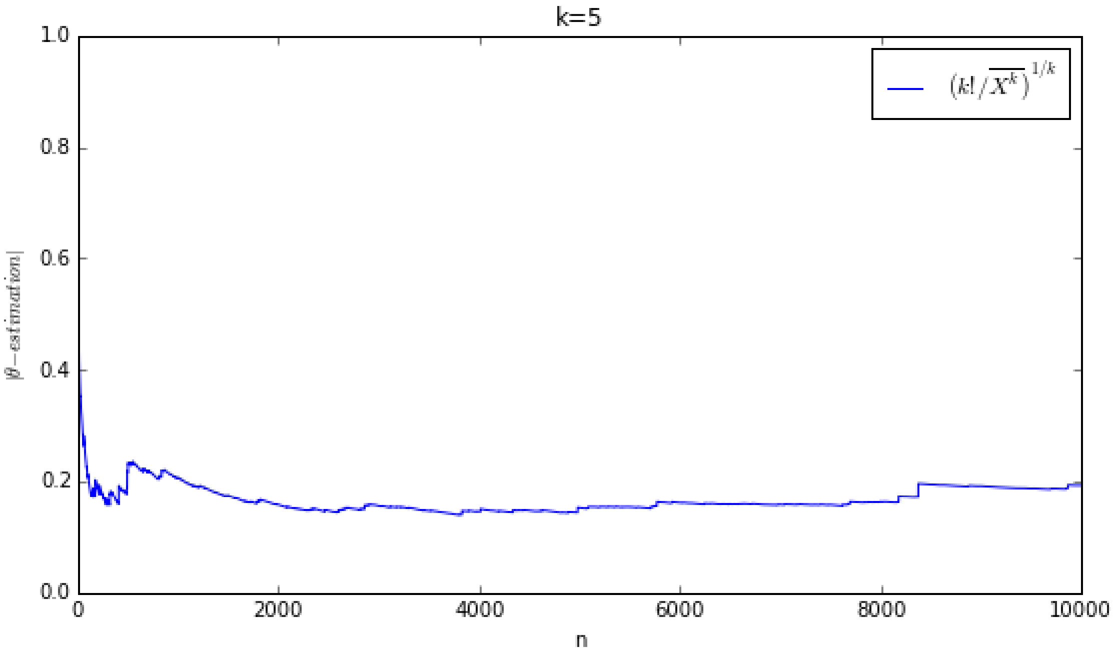
```
Delta: 0.36472600718
```



```
Delta: 0.272950038274
```



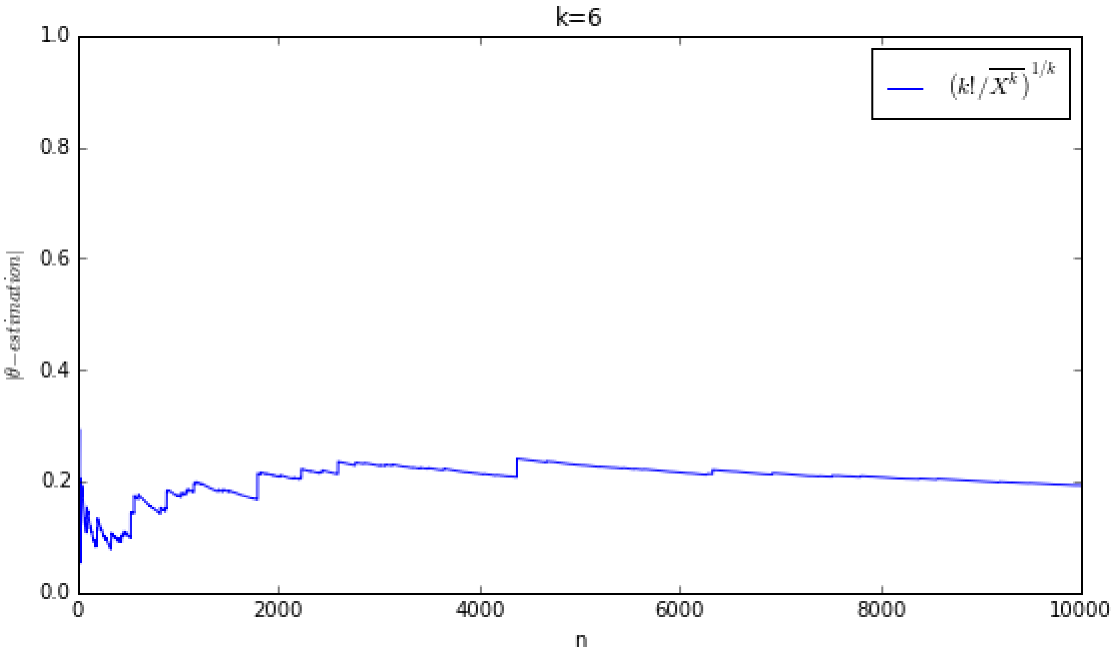
```
Delta: 0.215908474425
```



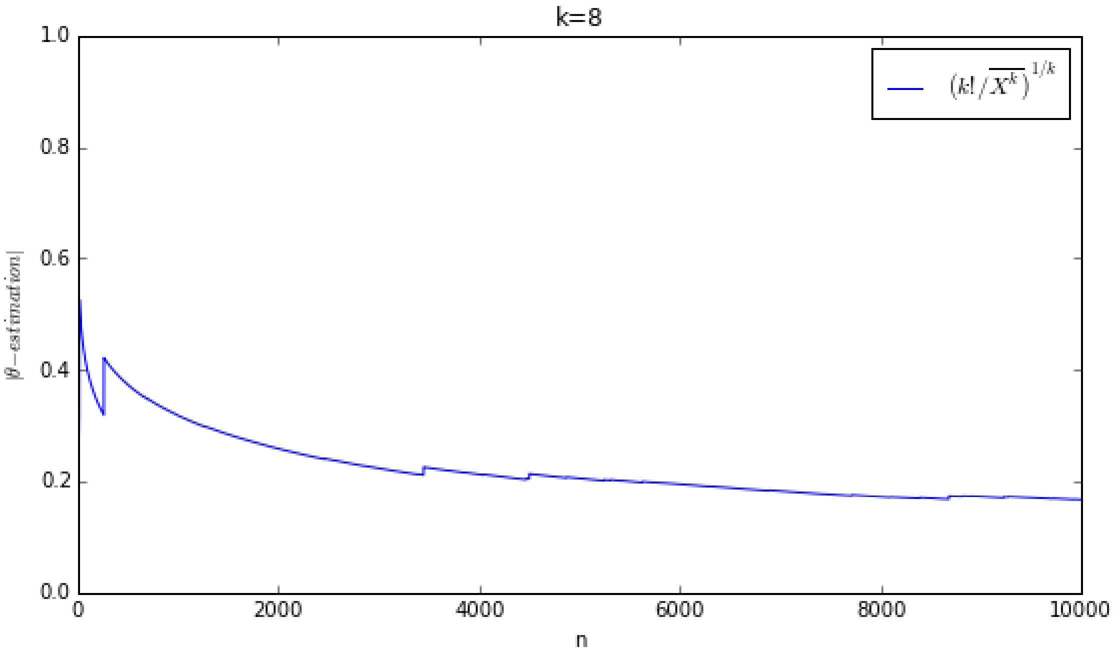
```
Delta: 0.19310100328
```

In [175]:

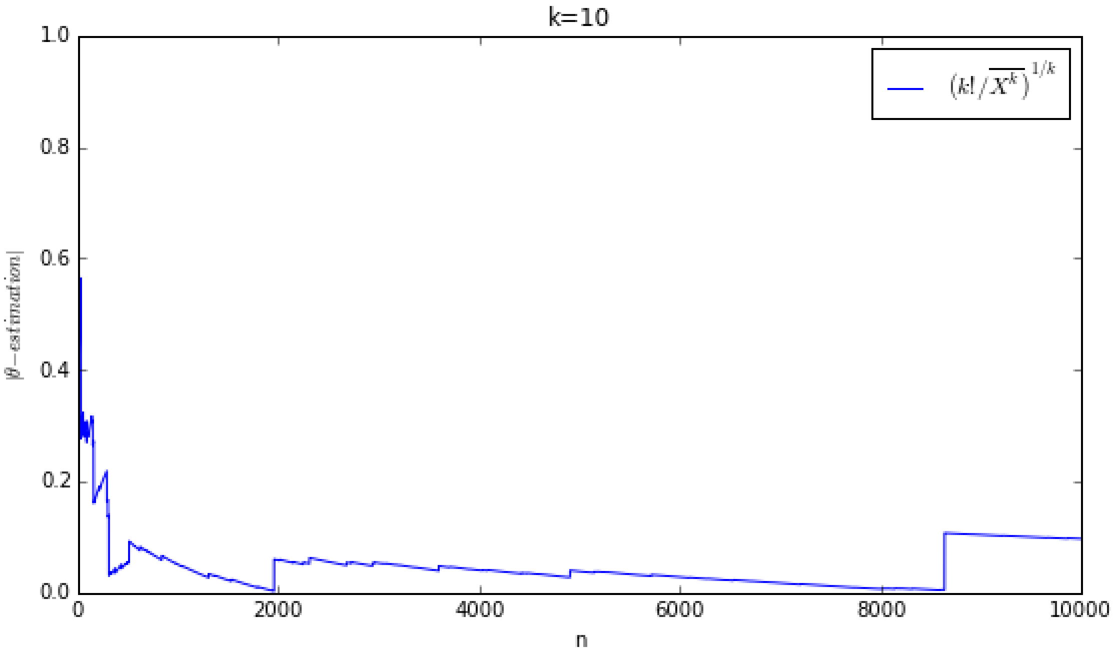
```
for k in range(6, 15, 2):
    showGraphs(k, 1)
```



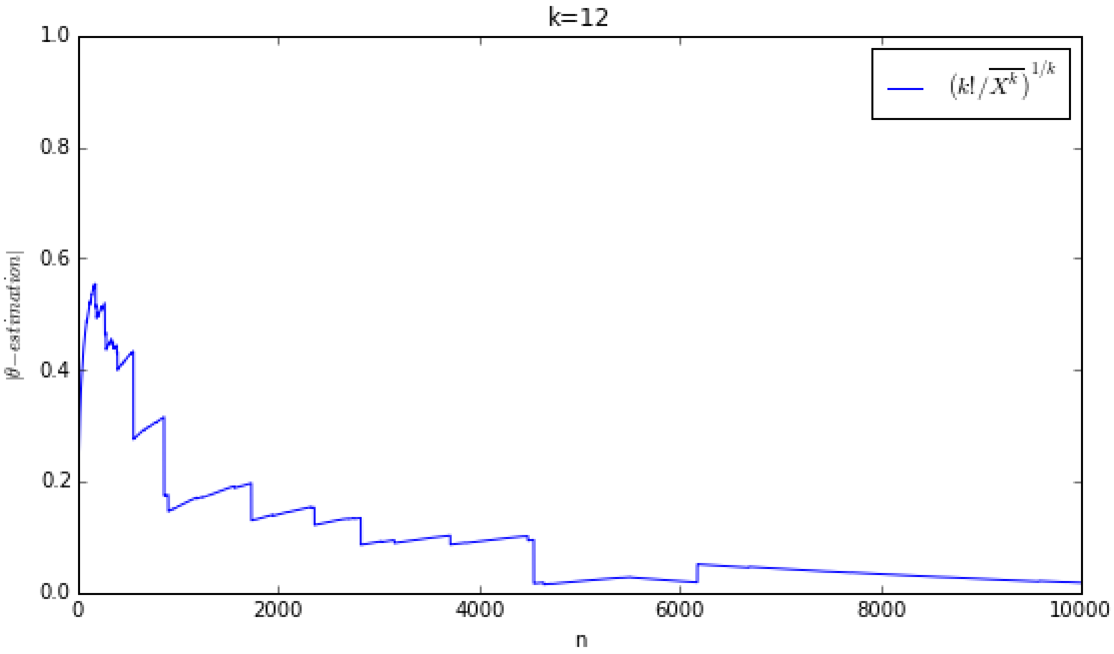
```
Delta: 0.192901557879
```



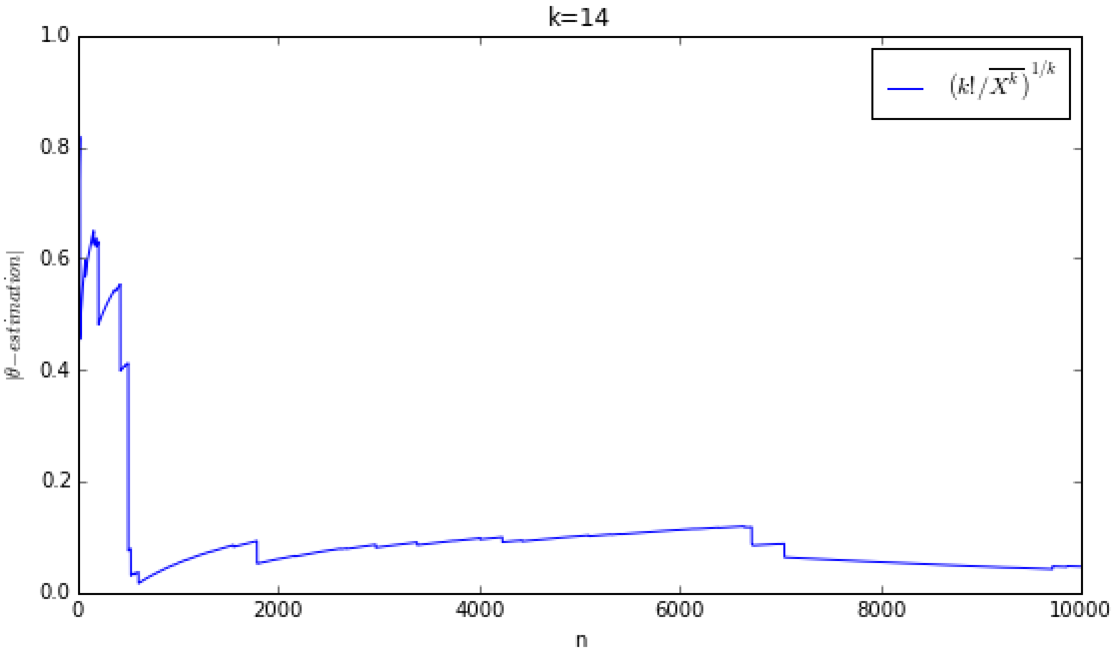
```
Delta: 0.166687686614
```



```
Delta: 0.0965245990554
```



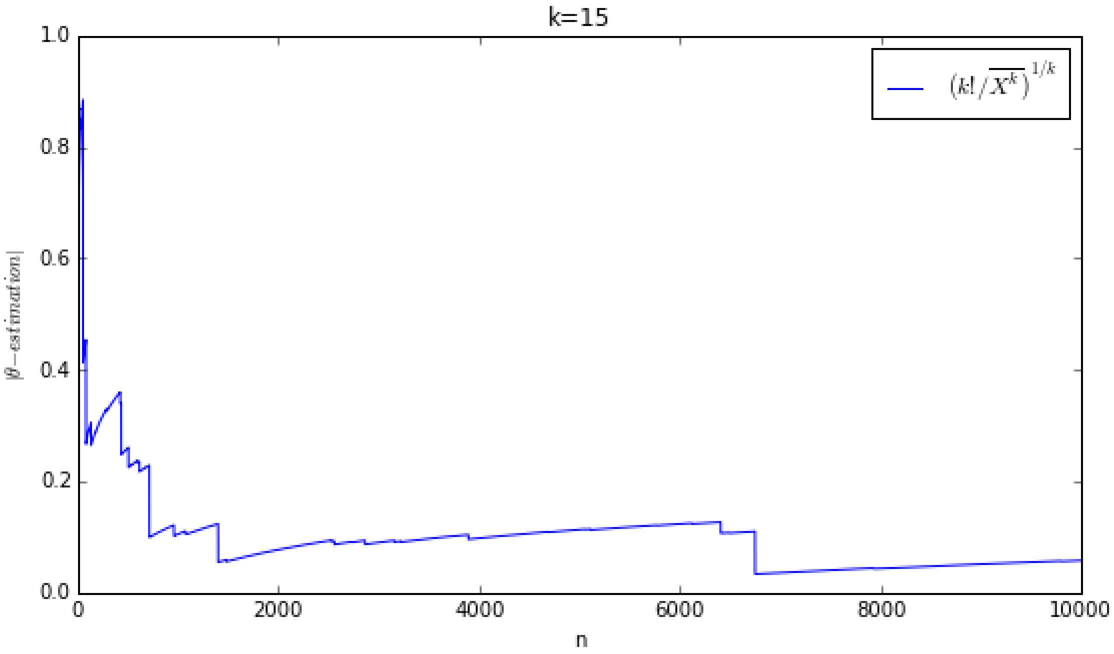
```
Delta: 0.0177752180262
```



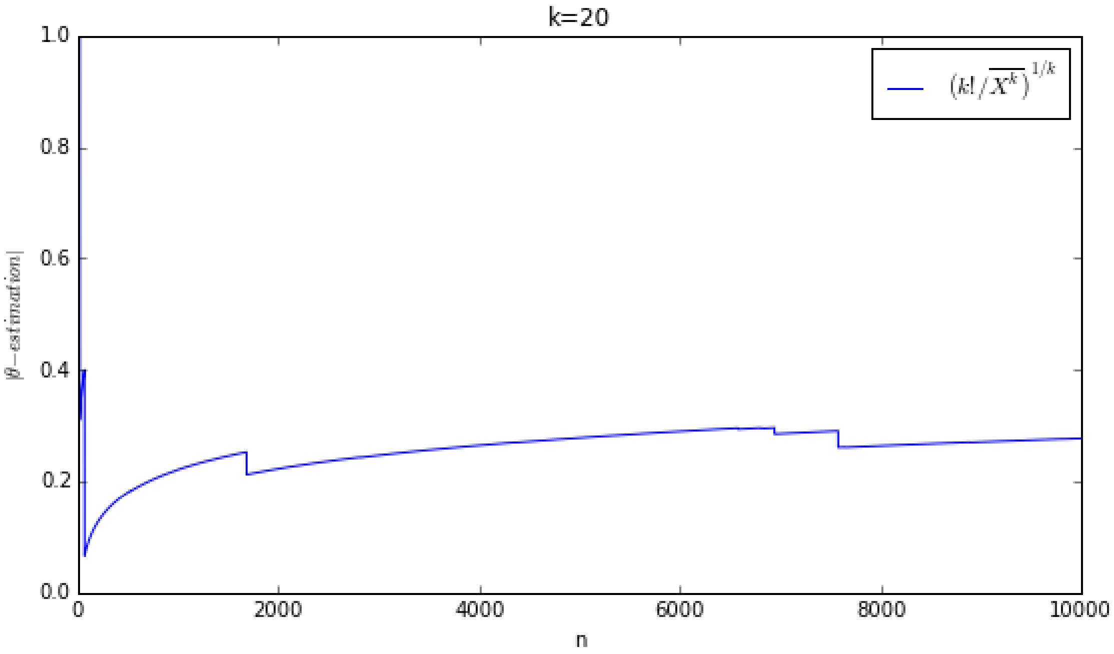
```
Delta: 0.0467209748597
```

In [177]:

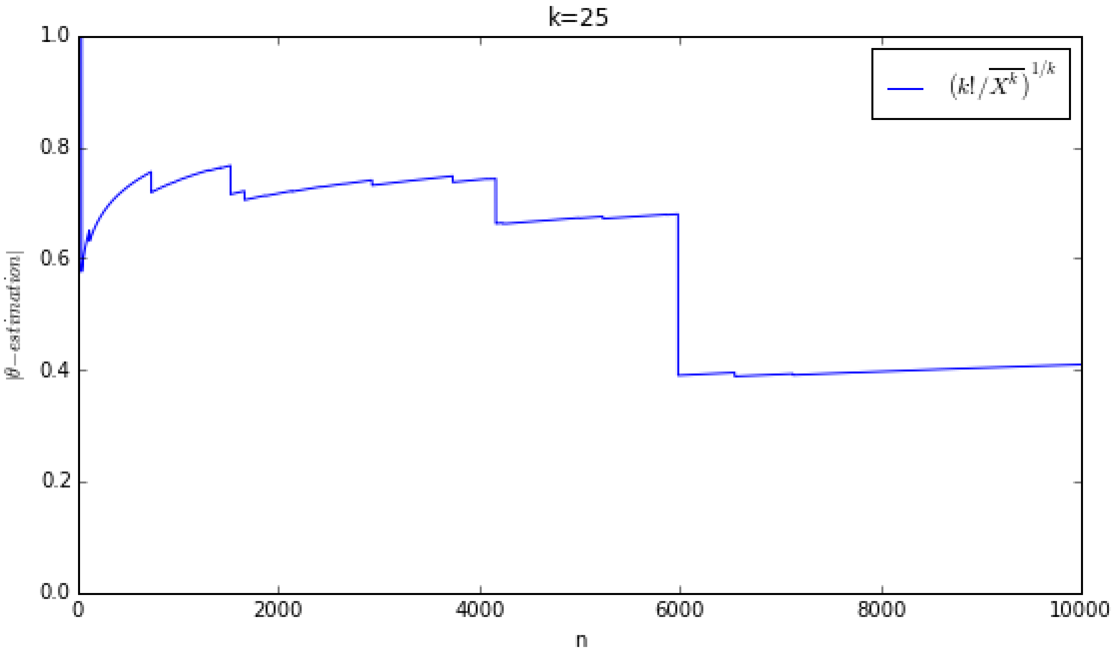
```
for k in range(15, 36, 5):
    showGraphs(k, 1)
```



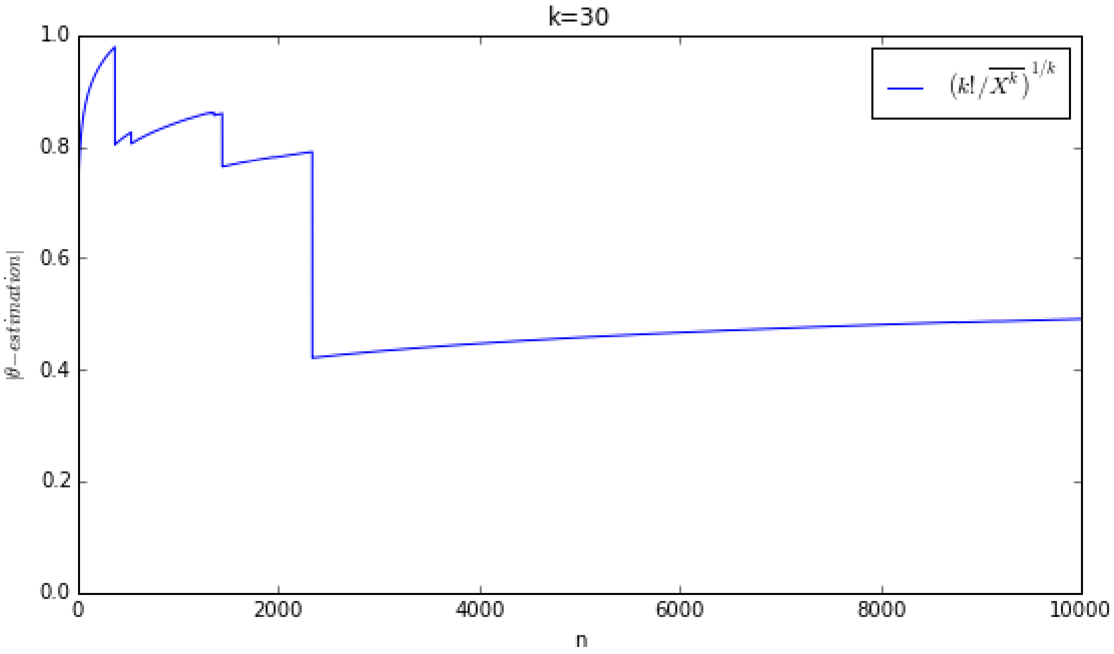
```
Delta: 0.0574604820669
```



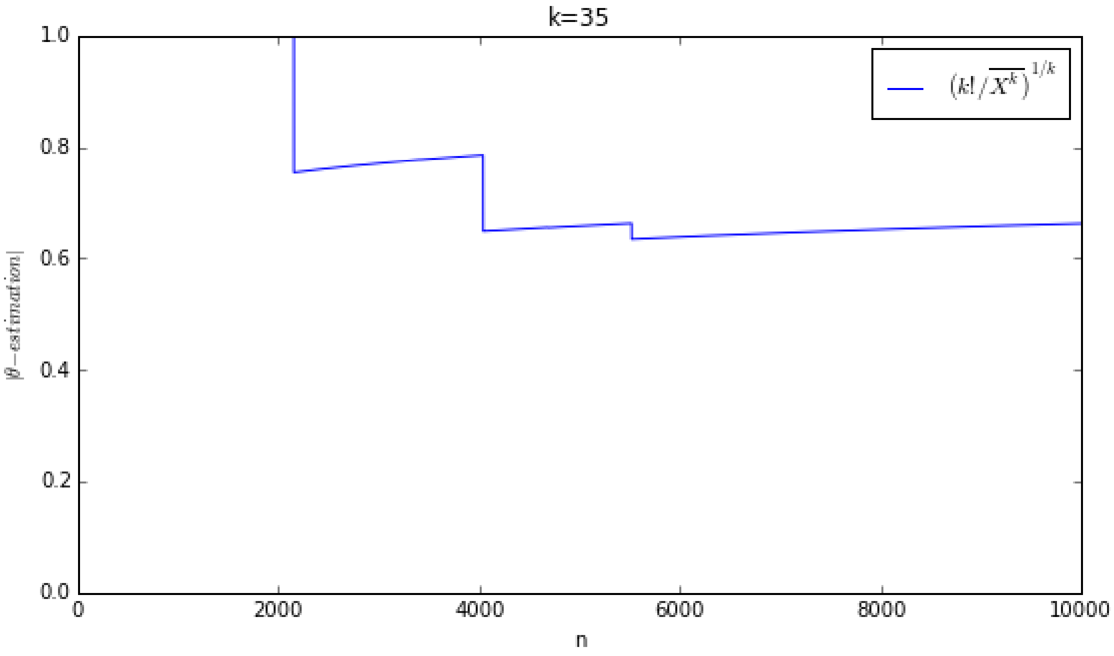
```
Delta: 0.277355612366
```



```
Delta: 0.409570745201
```



```
Delta: 0.490711252295
```



```
Delta: 0.662282166018
```

Лучшая оценка получилась при k = 12

```
In [ ]:
```