In [1]:
```
%matplotlib inline

import matplotlib.pyplot as plt
import numpy as np
from scipy.stats import norm
```

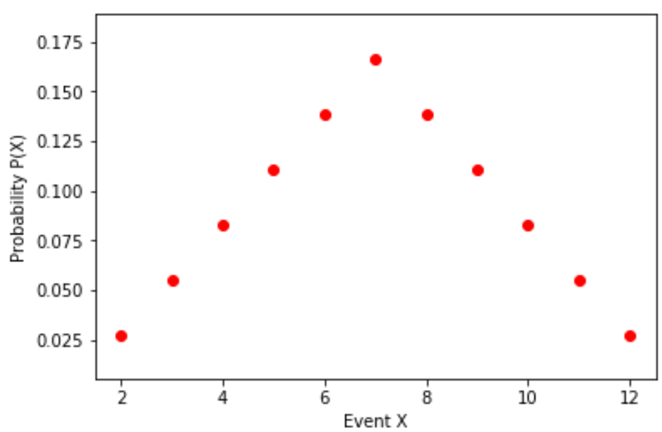
Question 1

In [2]:
```
die1 = [1,2,3,4,5,6]
die2 = [1,2,3,4,5,6]

x_px = dict()
total_events = len(die1)*len(die2)

for i in die1:
    for j in die2:
        sum_dice = i+j
        if sum_dice in x_px.keys():
            x_px[sum_dice] += 1/total_events
        else:
            x_px[sum_dice] = 1/total_events
```

Question 1 (a)

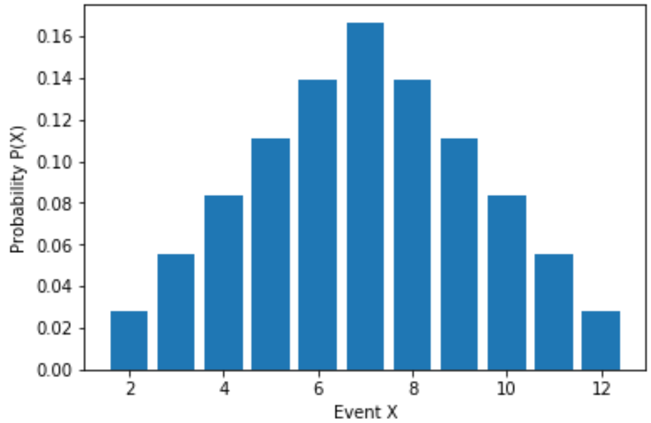
In [3]:
```
plt.scatter(list(x_px.keys()), list(x_px.values()), color="red")
plt.xlabel("Event X")
plt.ylabel("Probability P(X)")
```

Out[3]:
```
Text(0, 0.5, 'Probability P(X)')
```



In [4]:
```
plt.bar(x_px.keys(), x_px.values())
plt.xlabel("Event X")
plt.ylabel("Probability P(X)")
```

Out[4]:
```
Text(0, 0.5, 'Probability P(X)')
```



Question 1 (b) The distribution looks like normal distribution. It has a bell-curve like distribution, centered at X=7

Question 1(c)

In [5]:
```
print("Pr(X=8 or X=9) = ",x_px[8] + x_px[9])
```

```
Pr(X=8 or X=9) =  0.25
```

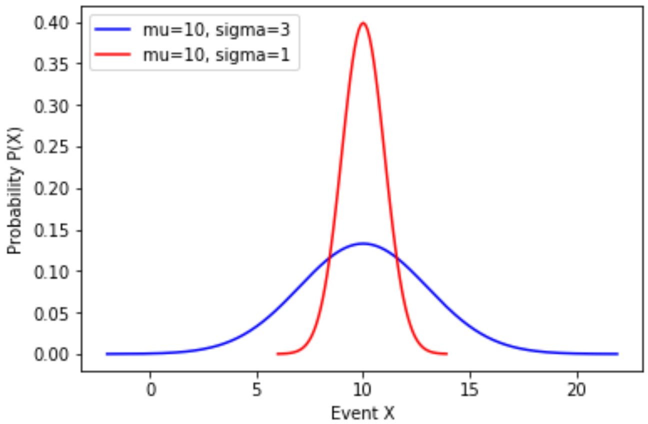
Question 2

Question 2 (a)

In [5]:
```
total_events = 100000
dx = 0.1

### mu=10, sigma=3

mu1, sigma1 = 10, 3
x1 = np.arange(mu1 - 4*sigma1, mu1 + 4*sigma1, dx)
px1 = norm.pdf(x1, mu1, sigma1)

x_px1 = dict(zip(x1, px1))

### mu=10, sigma=1

mu2, sigma2 = 10, 1
x2 = np.arange(mu2 - 4*sigma2, mu2 + 4*sigma2, dx)
px2 = norm.pdf(x2, mu2, sigma2)

x_px2 = dict(zip(x2, px2))
```

In [6]:
```
plt.plot(x1, px1, color='blue')
plt.plot(x2, px2, color='red')
plt.xlabel("Event X")
plt.ylabel("Probability P(X)")
plt.legend(['mu=10, sigma=3', 'mu=10, sigma=1'], loc='upper left')
```

Out[6]: <matplotlib.legend.Legend at 0x7fef2406a4e0>



Question 2 (b)

In [7]:
```
print("mu=10, sigma=3, Pr(X>11) = ", 1-norm.cdf(11, mu1, sigma1))
print("mu=10, sigma=1, Pr(X>11) = ", 1-norm.cdf(11, mu2, sigma2))
```

```
mu=10, sigma=3, Pr(X>11) =  0.369441340182
mu=10, sigma=1, Pr(X>11) =  0.158655253931
```

Question 2 (c)

In [8]:
```
print("Using arrays created earlier:")

print("mu=10, sigma=3, Pr(X>11) = ", np.sum([x_px1[i]*dx for i in x_px1.keys() if
i>11]))
print("mu=10, sigma=1, Pr(X>11) = ", np.sum([x_px2[i]*dx for i in x_px2.keys() if
i>11]))
```

```
Using arrays created earlier:
mu=10, sigma=3, Pr(X>11) =  0.375708757597
mu=10, sigma=1, Pr(X>11) =  0.146719619349
```

Question 3

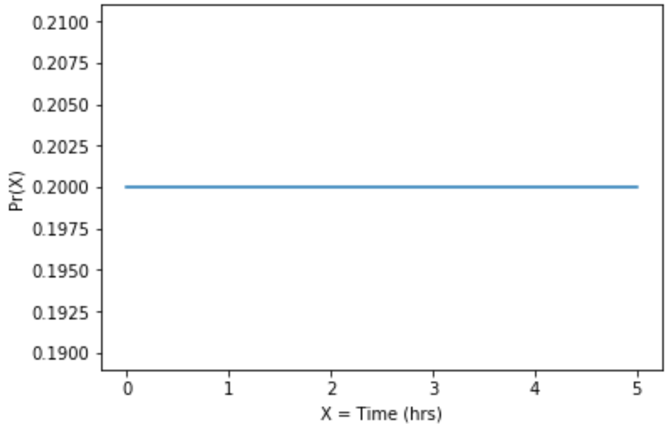
Question 3 (a)

In [26]:
```
times = [0,1,2,3,4,5]
pr_eq = [1/5]*len(times)
```

Out[26]: 6

In [23]:
```
plt.plot(times, pr_eq)
plt.xlabel("X = Time (hrs)")
plt.ylabel("Pr(X)")
```

Out[23]:
```
Text(0, 0.5, 'Pr(X)')
```



Question 3 (b)

In [28]:
```
print("E[X] = ", 1/2*(0+5))
```

```
E[X] =  2.5
```

Question 3 (c)

In [3]:
```
print("Pr (Earthquake between midnight and 1:30am) = ", 1.5/5)
```

```
Pr (Earthquake between midnight and 1:30am) =  0.3
```

Question 4

a) P(HHHH) = 1/16

b) P(HTHT) = 1/16

c) 1/2 * 1/2 = 1/4

d) 3/4

e) 1 - P(TTTT) = 5/16

Question 7

P(dis) = .005

P(no dis) = .995

P(pos | dis) = .99

P(pos | no dis) = .01

P(pos) = .99 *.005 + .01* .995 = .0149

P(dis | pos) = .99*.005 / .0149 = .33

In [ ]: