In [ ]:
```
# import needed libries
import numpy as np
import matplotlib.pyplot as plt
```

In [ ]:
```
# generating possion process with lambda > 0, up untile time T
def PoissonProcess(lam,T):
    timeCount = 0  # the total time that has elapsed after each arrival
    val = 0  # value of each poisson process
    Nt = [0]   # a list of values of poisson process
    times = [0]  # a list that contains the arrival times
    while(timeCount <= T):
        x = np.random.exponential(1/lam)
        timeCount = timeCount + x
        if(timeCount <= T):
            val = val + 1
            Nt.append(val)
            times.append(timeCount)
        if(timeCount > T):
            Nt.append(val)
            times.append(T)
    return Nt, times
```

In [8]:
```
PoissonSim = PoissonProcess(1,10)  # simulate poisson process with lambda = 1 on the time interval [0,1]
PoissonSim
```

Out[8]:
```
([0, 1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 10],
 [0,
  0.9943622635361881,
  1.1849606963521522,
  2.045322987128204,
  4.1176100304837355,
  4.672159794885869,
  5.651773326807148,
  5.703975083938087,
  6.244097332688594,
  8.865803777281666,
  9.9093738178172,
  10])
```

In [ ]:
```
# draw the poisson process
Nt = PoissonSim[0]
times = PoissonSim[1]
plt.plot(times, Nt, drawstyle='steps-post')
```

In [5]:
```
# generating a compound possion process with lambda > 0, up untile time T
def CompoundPoissonProcess(lam,T,mu,sigma):
    timeCount = 0
    val = 0
    Nt = [0]
    times = [0]
    while(timeCount <= T):
        x = np.random.exponential(1/lam)
        timeCount = timeCount + x
        if(timeCount <= T):
            Z = np.random.normal(mu, sigma)
            val = val + np.exp(Z)
            Nt.append(val)
            times.append(timeCount)
        if(timeCount > T):
            Nt.append(val)
            times.append(T)
    return Nt, times
```

In [ ]:
```
# plot the compound poisson process with lamda = 1,  mu = 0, sigma = 1 on the time interval [0,10]
CompoundPoissonSim = CompoundPoissonProcess(1,10,0,1)
CNt = CompoundPoissonSim[0]
Ctimes = CompoundPoissonSim[1]
plt.plot(Ctimes, CNt, drawstyle='steps-post')
```