```
import numpy as np
data = [1,4,2,5,5,10,6,12,7,15]
data = np.array(data)
data = data.reshape(5,2)

##function
def pearsons_correlation_coeddicient_and_covariance(data):
    meanx = 0
    meany = 0
    varx = 0
    vary = 0
    covxy = 0
    for i in range(data.shape[0]):
        meanx += float(data[i][0])
        meany += float(data[i][1])
    meanx = meanx/data.shape[0]
    meany = meany/data.shape[0]
    for i in range(data.shape[0]):
        varx +=    (float(data[i][0])-meanx)**2
        vary +=    (float(data[i][1])-meany)**2
    for i in range(data.shape[0]):
        covxy += (float(data[i][0])-meanx)*(float(data[i][1])-meany)
    pearsons_correlations = covxy/((varx**0.5) * (vary**0.5))
    return meanx,meany,varx,vary,covxy,pearsons_correlations

##pearsons_correlations
meanx,meany,varx,vary,covxy,pearsons_correlations =
pearsons_correlation_coeddicient_and_covariance(data)
print(pearsons_correlations)
####output :0.0.9910615723046898


##target data
import numpy as np
np.random.seed(1)
n=100
noise = (np.random.rand(n)-0.5)*2
x = np.random.normal(0,1,n)
```

```
y = 5 * x    +    10 * noise

##data in list
datalist = []
for i in range(n):
        datalist.append(float(x[i]))
        datalist.append(float(y[i]))
datalist = np.array(datalist)
datalist = datalist.reshape(n,2)

##pearsons_correlations
meanx,meany,varx,vary,covxy,pearsons_correlations =
pearsons_correlation_coeddicient_and_covariance(datalist)
print(pearsons_correlations)
####output :0.5530828259342412
```