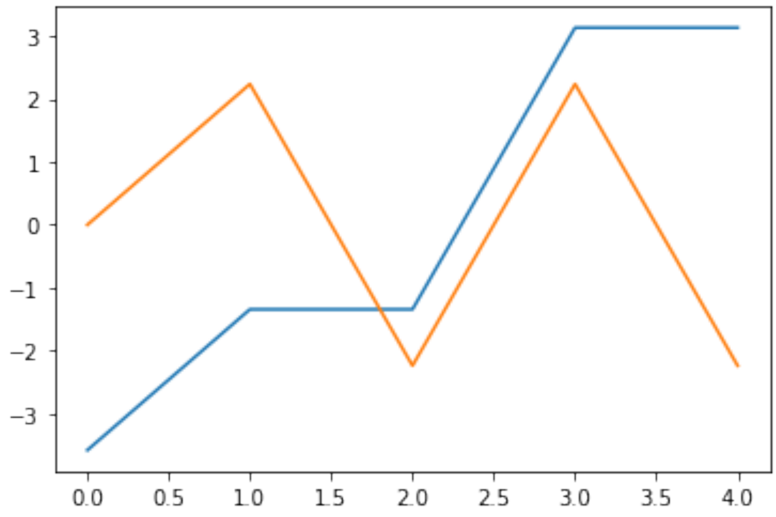
M.Nilofer Sultana,AP17110010058.

Implementation of PCA

In [10]:
```
from numpy import array
from numpy import mean
from numpy import cov
from numpy.linalg import eig
from matplotlib import pyplot as plt
# define a matrix
A = array([[2, 1], [3, 4], [5, 0],[7, 6],[9, 2]])
print("The given matrix\n:",A)
# calculate the mean of each column
M = mean(A.T, axis=1)
print("the mean is :",M)
# step 1 is standatisation the main aim is to make the data to zero me
C = A - M
print("the standardised matrix is: ", C)
# step 2 is to calculate covariance matrix of the standardised matrix.
V = cov(C.T)
print("the covariance matrix is: ", V)
# step 3 is to calculate eigen values and eigen vecors of the covariar
values, vectors = eig(V)
print("the eigen vectors: ",vectors)
print("the eigen values are: ",values)
# project data
P = vectors.T.dot(C.T)
print("the final reduced matrix: ",P.T)
plt.plot(P.T)
plt.show()
```

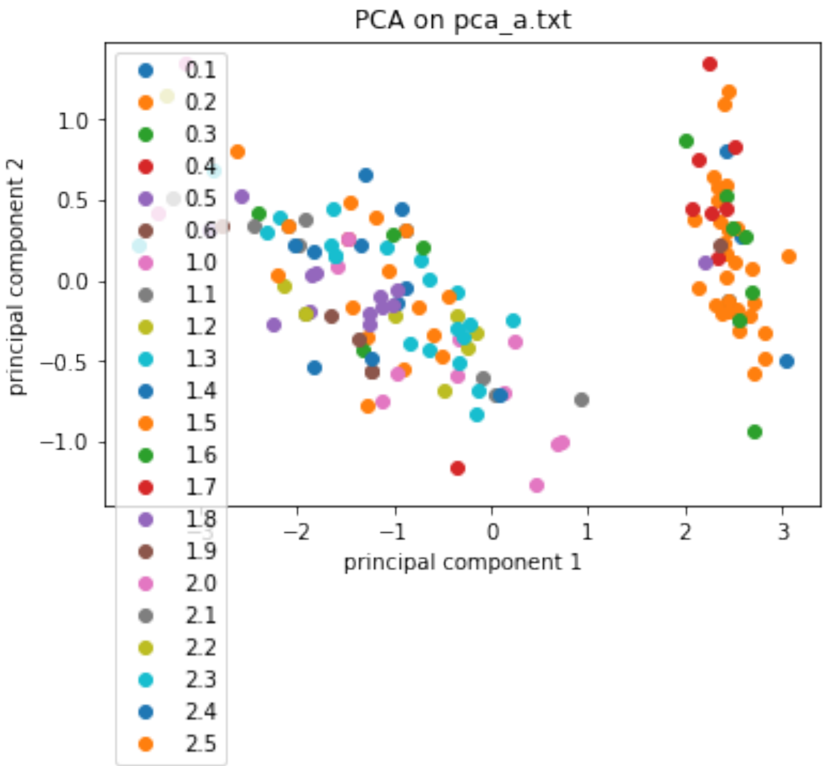
```
The given matrix
: [[2 1]
 [3 4]
 [5 0]
 [7 6]
 [9 2]]
the mean is : [5.2 2.6]
the standardised matrix is:  [[-3.2 -1.6]
 [-2.2  1.4]
 [-0.2 -2.6]
 [ 1.8  3.4]
 [ 3.8 -0.6]]
the covariance matrix is:  [[8.2 1.6]
 [1.6 5.8]]
the eigen vectors:  [[ 0.89442719 -0.4472136 ]
 [ 0.4472136   0.89442719]]
the eigen values are:  [9. 5.]
the final reduced matrix:  [[-3.57770876e+00 -2.22044605e-16]
```

```
[-1.34164079e+00  2.23606798e+00]
[-1.34164079e+00 -2.23606798e+00]
[ 3.13049517e+00  2.23606798e+00]
[ 3.13049517e+00 -2.23606798e+00]]
```



In [12]:
```
import numpy as np
import matplotlib.pyplot as plt
import pandas as pd
from sklearn import datasets
iris=datasets.load_iris()
x=iris.data
y=iris.target
df=pd.DataFrame(x)
arr=np.array(df)
df.head(n=10)
class_label = pd.DataFrame(df.iloc[:,-1])
class_label.columns = ['label']
df = df.iloc[:, :-1]
df = df.sub(df.mean(axis=0), axis=1)
df_mat = np.asmatrix(df)
sigma = np.cov(df_mat.T)
eigVals, eigVec = np.linalg.eig(sigma)
sorted_index = eigVals.argsort()[::-1]
eigVals = eigVals[sorted_index]
eigVec = eigVec[:,sorted_index]
eigVec = eigVec[:,:2]
transformed = df_mat.dot(eigVec)
#horizontally stack transformed data set with class label.
final_df = np.hstack((transformed, class_label))
#convert the numpy array to data frame
final_df = pd.DataFrame(final_df)
#define the column names
final_df.columns = ['x','y','label']
groups = final_df.groupby('label')
figure, axes = plt.subplots()
axes.margins(0.05)
for name, group in groups:
    axes.plot(group.x, group.y, marker='o', linestyle='', ms=6, label=
```

```
    axes.set_title("PCA on pca_a.txt")
axes.legend()
plt.xlabel("principal component 1")
plt.ylabel("principal component 2")
plt.show()
```



In [ ]: