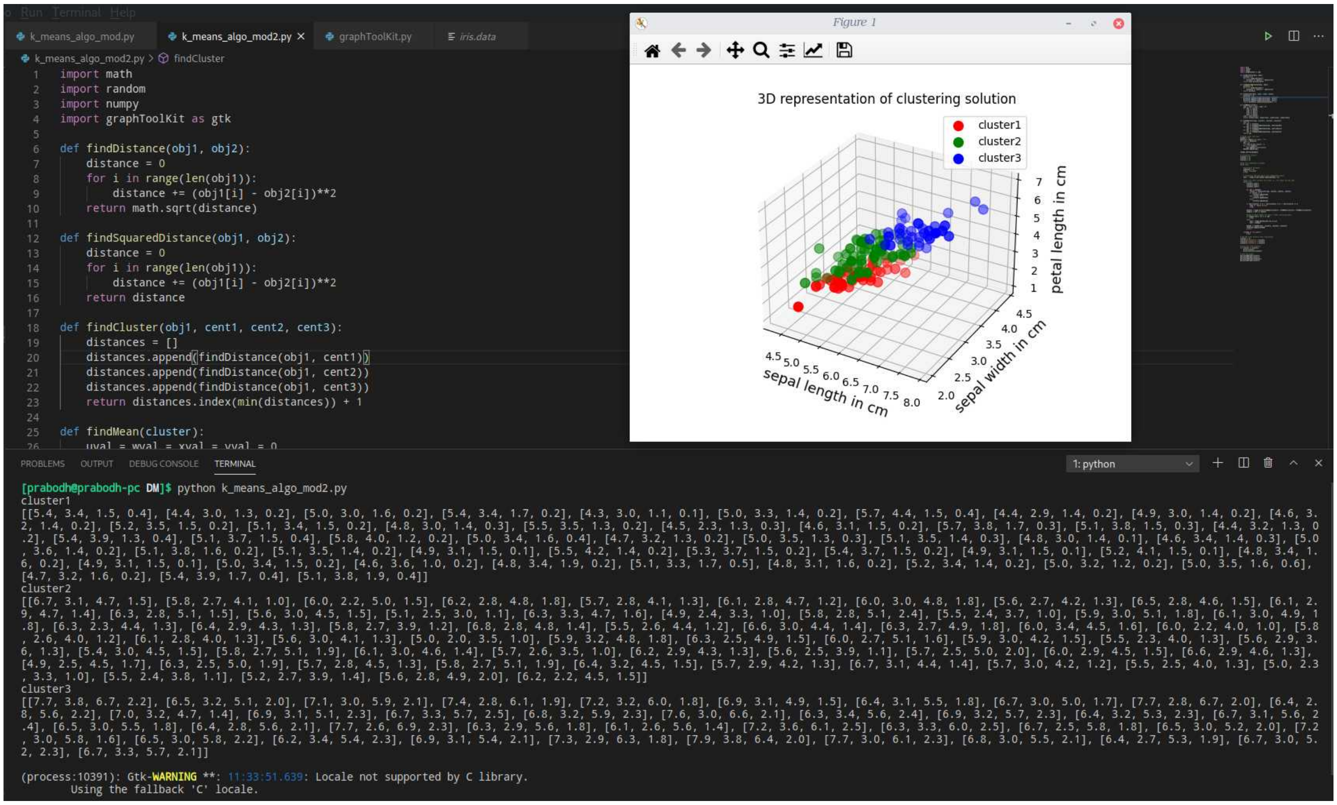
graphToolKit.py

```
import matplotlib.pyplot as plt
from mpl_toolkits.mplot3d import Axes3D
import numpy


def plot4DGraph(clusters):

    fig = plt.figure()
    ax = fig.add_subplot(111, projection='3d')

    iter = 0
    for cluster in clusters:
        u_val = [obj[0] for obj in clusters[cluster]]
        v_val = [obj[1] for obj in clusters[cluster]]
        w_val = [obj[2] for obj in clusters[cluster]]
        x_val = [obj[3] for obj in clusters[cluster]]

        if iter == 0:
            img1 = ax.scatter(u_val, v_val, w_val, c = x_val, cmap = plt.winter(), label = 'cluster1')
            cbar = fig.colorbar(img1, shrink = 0.5, aspect = 10)
        elif iter == 1:
            img2 = ax.scatter(u_val, v_val, w_val, c = x_val, cmap = plt.spring(), label = 'cluster2')
            cbar = fig.colorbar(img2, shrink = 0.5, aspect = 10)
        else:
            img3 = ax.scatter(u_val, v_val, w_val, c = x_val, cmap = plt.gray(), label = 'cluster3')
            cbar = fig.colorbar(img3, shrink = 0.5, aspect = 10)

        iter += 1
        cbar.ax.get_yaxis().labelpad = 15
        cbar.ax.set_ylabel('petal width in cm')
        cbar.ax.get_xaxis().labelpad = 15
        cbar.ax.set_xlabel('cluster' + str(iter))

    ax.set_xlabel('sepal length in cm', rotation=150)
    ax.set_ylabel('sepal width in cm')
    ax.set_zlabel(r'petal length in cm', rotation=60)


    plt.title("4D representation of clustering solution")
    plt.show()

def plot3DGraph(clusters):

    fig = plt.figure()
    ax = fig.add_subplot(111, projection='3d')
    colorArray = ['red', 'green', 'blue']
    iter = 0

    for cluster in clusters:
        u_val = [obj[0] for obj in clusters[cluster]]
        v_val = [obj[1] for obj in clusters[cluster]]
        w_val = [obj[2] for obj in clusters[cluster]]

        ax.scatter(u_val, v_val, w_val, s = 75, c = colorArray[iter], label = 'cluster' + str(iter + 1))
        iter += 1
    plt.legend()
    ax.set_xlabel('sepal length in cm', fontsize=13, rotation=150)
    ax.set_ylabel('sepal width in cm', fontsize=13)
    ax.set_zlabel(r'petal length in cm', fontsize=13, rotation=60)
    plt.title("3D representation of clustering solution")
    plt.show()

def plotSSEGraph(sseValues):

    x_val = numpy.arange(1,151,1)
    y_val = sseValues
    plt.plot(x_val, y_val)
    plt.scatter(x_val, y_val, c = "red", marker= '+', label = "round 1")

    plt.xlabel("iteration")
    plt.ylabel("SSE values")
    plt.title("iteration vs SSE values")
    plt.grid()
    plt.legend()
    plt.show()

def plot2DGraph(clusters):

    colorArray = ['red', 'green', 'blue']
    attributes = ["sepal length", "sepal width", "petal length", "petal width"]
    for i in range(0,3,2):
        iter = 0
        for cluster in clusters:
            u_val = [obj[0 + i] for obj in clusters[cluster]]
            v_val = [obj[1 + i] for obj in clusters[cluster]]

            plt.scatter(u_val, v_val, s = 50, c = colorArray[iter], label = "cluster" + str(iter + 1))
            iter += 1

        plt.xlabel(attributes[0 + i] + "(cm)", fontsize = 15)
        plt.ylabel(attributes[1 + i] + "(cm)", fontsize = 15)
        plt.title(attributes[0 + i] + " vs " + attributes[1 + i] + " of clusters", fontsize = 20)
        plt.show()
```

k_mean_algo_mod.py

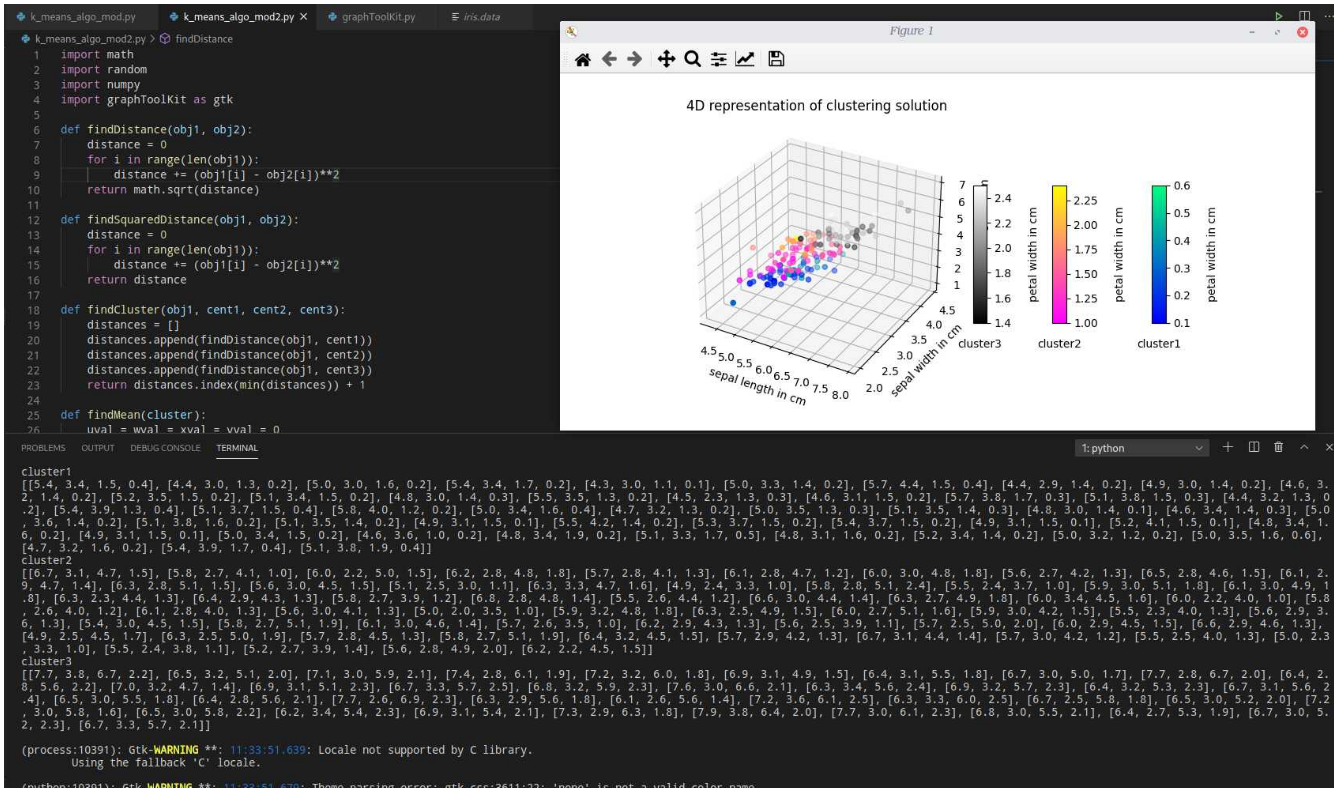
```
import math
import random
import numpy
import graphToolKit as gtk

def findDistance(obj1, obj2):
    distance = 0
    for i in range(len(obj1)):
        distance += (obj1[i] - obj2[i])**2
    return math.sqrt(distance)

def findSquaredDistance(obj1, obj2):
    distance = 0
    for i in range(len(obj1)):
        distance += (obj1[i] - obj2[i])**2
    return distance

def findCluster(obj1, cent1, cent2, cent3):
    distances = []
    distances.append(findDistance(obj1, cent1))
    distances.append(findDistance(obj1, cent2))
    distances.append(findDistance(obj1, cent3))
    return distances.index(min(distances)) + 1

def findMean(cluster):
    uval = wval = xval = yval = 0
    for obj in cluster:
        uval += obj[0]
        wval += obj[1]
        xval += obj[2]
        yval += obj[3]
    size = len(cluster)
    return [(uval/size), (wval/size), (xval/size), (yval/size)]

def findSSE(centroids, cluster1, cluster2, cluster3):
    sse = 0
    for obj in cluster1:
        sse += findSquaredDistance(obj, centroids[0])
    for obj in cluster2:
        sse += findSquaredDistance(obj, centroids[1])
    for obj in cluster3:
        sse += findSquaredDistance(obj, centroids[2])
    return sse

# taking input from file
dataSet = []
dataFile = open("iris.data", "r")
for line in dataFile:
    obj = []
    x = line.strip().split(",")
    for i in range(4):
        obj.append((float)(x[i]))
    dataSet.append(obj)

random.shuffle(dataSet)

# initialise clusters
cluster1 = []
cluster2 = []
cluster3 = []

#loop till clusering is success
while True:

    #initialise variables
    sseValues = []
    flag = "all_good"
    i = 0

    # initialize centroid values with random data points
    cent =  numpy.array(random.sample(dataSet, 3))

    #loop till final clusters are found i.e., till means are the same
    while True:
        cluster1.clear()
        cluster2.clear()
        cluster3.clear()

        for obj in dataSet:
            cluster = findCluster(obj, cent[0], cent[1], cent[2])
            if cluster == 1:
                cluster1.append(obj)
            elif cluster == 2:
                cluster2.append(obj)
            else:
                cluster3.append(obj)

        if len(cluster1) == 0 or len(cluster2) == 0 or len(cluster3) == 0:
            flag == "empty_cluster"
            break

        newCent = numpy.array([findMean(cluster1), findMean(cluster2), findMean(cluster3)])
        compare = cent == newCent

        #break of means remain the same => final clustering found
        if compare.all() and i >= 150:
            break
        else:
            cent = numpy.delete(cent,[0,1,2],0)
            cent = newCent

        newSSE = findSSE(cent, cluster1, cluster2, cluster3)
        sseValues.append(newSSE)
        i += 1

    if(flag == "all_good"):
        break

# add the final clusters into a dictionary
clusters = {}
clusters["cluster1"] = cluster1
clusters["cluster2"] = cluster2
clusters["cluster3"] = cluster3

# print the final clusters
for cluster in clusters:
    print(cluster)
    print(clusters[cluster])

# plot the graphs
gtk.plot3DGraph(clusters)
gtk.plot4DGraph(clusters)
gtk.plotSSEGraph(sseValues)
gtk.plot2DGraph(clusters)
```

Screenshots



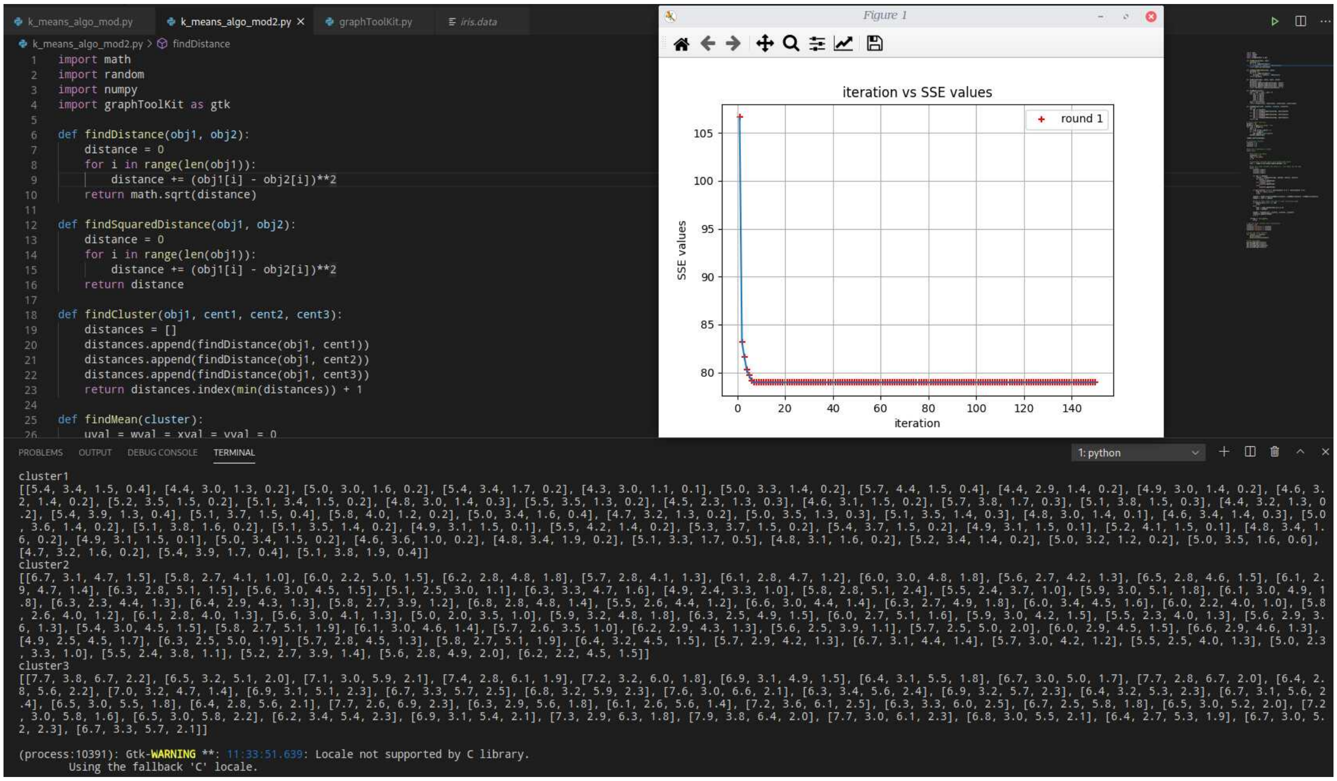
```python
import math
import random
import numpy
import graphToolKit as gtk

def findDistance(obj1, obj2):
    distance = 0
    for i in range(len(obj1)):
        distance += (obj1[i] - obj2[i])**2
    return math.sqrt(distance)

def findSquaredDistance(obj1, obj2):
    distance = 0
    for i in range(len(obj1)):
        distance += (obj1[i] - obj2[i])**2
    return distance

def findCluster(obj1, cent1, cent2, cent3):
    distances = []
    distances.append(findDistance(obj1, cent1))
    distances.append(findDistance(obj1, cent2))
    distances.append(findDistance(obj1, cent3))
    return distances.index(min(distances)) + 1

def findMean(cluster):
    uval = wval = xval = yval = 0
```

```
[prabodh@prabodh-pc DM]$ python k_means_algo_mod2.py
cluster1
[[5.4, 3.4, 1.5, 0.4], [4.4, 3.0, 1.3, 0.2], [5.0, 3.0, 1.6, 0.2], [5.4, 3.4, 1.7, 0.2], [4.3, 3.0, 1.1, 0.1], [5.0, 3.3, 1.4, 0.2], [5.7, 4.4, 1.5, 0.4], [4.4, 2.9, 1.4, 0.2], [4.9, 3.0, 1.4, 0.2], [4.6, 3.
2, 1.4, 0.2], [5.2, 3.5, 1.5, 0.2], [5.1, 3.4, 1.5, 0.2], [4.8, 3.0, 1.4, 0.3], [5.5, 3.5, 1.3, 0.2], [4.5, 2.3, 1.3, 0.3], [4.6, 3.1, 1.5, 0.2], [5.7, 3.8, 1.7, 0.3], [5.1, 3.8, 1.5, 0.3], [4.4, 3.2, 1.3, 0
.2], [5.4, 3.9, 1.3, 0.4], [5.1, 3.7, 1.5, 0.4], [5.8, 4.0, 1.2, 0.2], [5.0, 3.4, 1.6, 0.4], [4.7, 3.2, 1.3, 0.2], [5.0, 3.5, 1.3, 0.3], [5.1, 3.5, 1.4, 0.3], [4.8, 3.0, 1.4, 0.1], [4.6, 3.4, 1.4, 0.3], [5.0
, 3.6, 1.4, 0.2], [5.1, 3.8, 1.6, 0.2], [5.1, 3.5, 1.4, 0.2], [4.9, 3.1, 1.5, 0.1], [5.5, 4.2, 1.4, 0.2], [5.3, 3.7, 1.5, 0.2], [5.4, 3.7, 1.5, 0.2], [4.9, 3.1, 1.5, 0.1], [5.2, 4.1, 1.5, 0.1], [4.8, 3.4, 1.
6, 0.2], [4.9, 3.1, 1.5, 0.1], [5.0, 3.4, 1.5, 0.2], [4.6, 3.6, 1.0, 0.2], [4.8, 3.4, 1.9, 0.2], [5.1, 3.3, 1.7, 0.5], [4.8, 3.1, 1.6, 0.2], [5.2, 3.4, 1.4, 0.2], [5.0, 3.2, 1.2, 0.2], [5.0, 3.5, 1.6, 0.6],
[4.7, 3.2, 1.6, 0.2], [5.4, 3.9, 1.7, 0.4], [5.1, 3.8, 1.9, 0.4]]
cluster2
[[6.7, 3.1, 4.7, 1.5], [5.8, 2.7, 4.1, 1.0], [6.0, 2.2, 5.0, 1.5], [6.2, 2.8, 4.8, 1.8], [5.7, 2.8, 4.1, 1.3], [6.1, 2.8, 4.7, 1.2], [6.0, 3.0, 4.8, 1.8], [5.6, 2.7, 4.2, 1.3], [6.5, 2.8, 4.6, 1.5], [6.1, 2.
9, 4.7, 1.4], [6.3, 2.8, 5.1, 1.5], [5.6, 3.0, 4.5, 1.5], [5.1, 2.5, 3.0, 1.1], [6.3, 3.3, 4.7, 1.6], [4.9, 2.4, 3.3, 1.0], [5.8, 2.8, 5.1, 2.4], [5.5, 2.4, 3.7, 1.0], [5.9, 3.0, 5.1, 1.8], [6.1, 3.0, 4.9, 1
.8], [6.3, 2.3, 4.4, 1.3], [6.4, 2.9, 4.3, 1.3], [5.8, 2.7, 3.9, 1.2], [6.8, 2.8, 4.8, 1.4], [5.5, 2.6, 4.4, 1.2], [6.6, 3.0, 4.4, 1.4], [6.3, 2.7, 4.9, 1.8], [6.0, 3.4, 4.5, 1.6], [6.0, 2.2, 4.0, 1.0], [5.8
, 2.6, 4.0, 1.2], [6.1, 2.8, 4.0, 1.3], [5.6, 3.0, 4.1, 1.3], [5.0, 2.0, 3.5, 1.0], [5.9, 3.2, 4.8, 1.8], [6.3, 2.5, 4.9, 1.5], [6.0, 2.7, 5.1, 1.6], [5.9, 3.0, 4.2, 1.5], [5.5, 2.3, 4.0, 1.3], [5.6, 2.9, 3.
6, 1.3], [5.4, 3.0, 4.5, 1.5], [5.8, 2.7, 5.1, 1.9], [6.1, 3.0, 4.6, 1.4], [5.7, 2.6, 3.5, 1.0], [6.2, 2.9, 4.3, 1.3], [5.6, 2.5, 3.9, 1.1], [5.7, 2.5, 5.0, 2.0], [6.0, 2.9, 4.5, 1.5], [6.6, 2.9, 4.6, 1.3],
[4.9, 2.5, 4.5, 1.7], [6.3, 2.5, 5.0, 1.9], [5.7, 2.8, 4.5, 1.3], [5.8, 2.7, 5.1, 1.9], [6.4, 3.2, 4.5, 1.5], [5.7, 2.9, 4.2, 1.3], [6.7, 3.1, 4.4, 1.4], [5.7, 3.0, 4.2, 1.2], [5.5, 2.5, 4.0, 1.3], [5.0, 2.3
, 3.3, 1.0], [5.5, 2.4, 3.8, 1.1], [5.2, 2.7, 3.9, 1.4], [5.6, 2.8, 4.9, 2.0], [6.2, 2.2, 4.5, 1.5]]
cluster3
[[7.7, 3.8, 6.7, 2.2], [6.5, 3.2, 5.1, 2.0], [7.1, 3.0, 5.9, 2.1], [7.4, 2.8, 6.1, 1.9], [7.2, 3.2, 6.0, 1.8], [6.9, 3.1, 4.9, 1.5], [6.4, 3.1, 5.5, 1.8], [6.7, 3.0, 5.0, 1.7], [7.7, 2.8, 6.7, 2.0], [6.4, 2.
8, 5.6, 2.2], [7.0, 3.2, 4.7, 1.4], [6.9, 3.1, 5.1, 2.3], [6.7, 3.3, 5.7, 2.5], [6.8, 3.2, 5.9, 2.3], [7.6, 3.0, 6.6, 2.1], [6.3, 3.4, 5.6, 2.4], [6.9, 3.2, 5.7, 2.3], [6.4, 3.2, 5.3, 2.3], [6.7, 3.1, 5.6, 2
.4], [6.5, 3.0, 5.5, 1.8], [6.4, 2.8, 5.6, 2.1], [7.7, 2.6, 6.9, 2.3], [6.3, 2.9, 5.6, 1.8], [6.1, 2.6, 5.6, 1.4], [7.2, 3.6, 6.1, 2.5], [6.3, 3.3, 6.0, 2.5], [6.7, 2.5, 5.8, 1.8], [6.5, 3.0, 5.2, 2.0], [7.2
, 3.0, 5.8, 1.6], [6.5, 3.0, 5.8, 2.2], [6.2, 3.4, 5.4, 2.3], [6.9, 3.1, 5.4, 2.1], [7.3, 2.9, 6.3, 1.8], [7.9, 3.8, 6.4, 2.0], [7.7, 3.0, 6.1, 2.3], [6.8, 3.0, 5.5, 2.1], [6.4, 2.7, 5.3, 1.9], [6.7, 3.0, 5.
2, 2.3], [6.7, 3.3, 5.7, 2.1]]

(process:10391): Gtk-WARNING **: 11:33:51.639: Locale not supported by C library.
        Using the fallback 'C' locale.
```



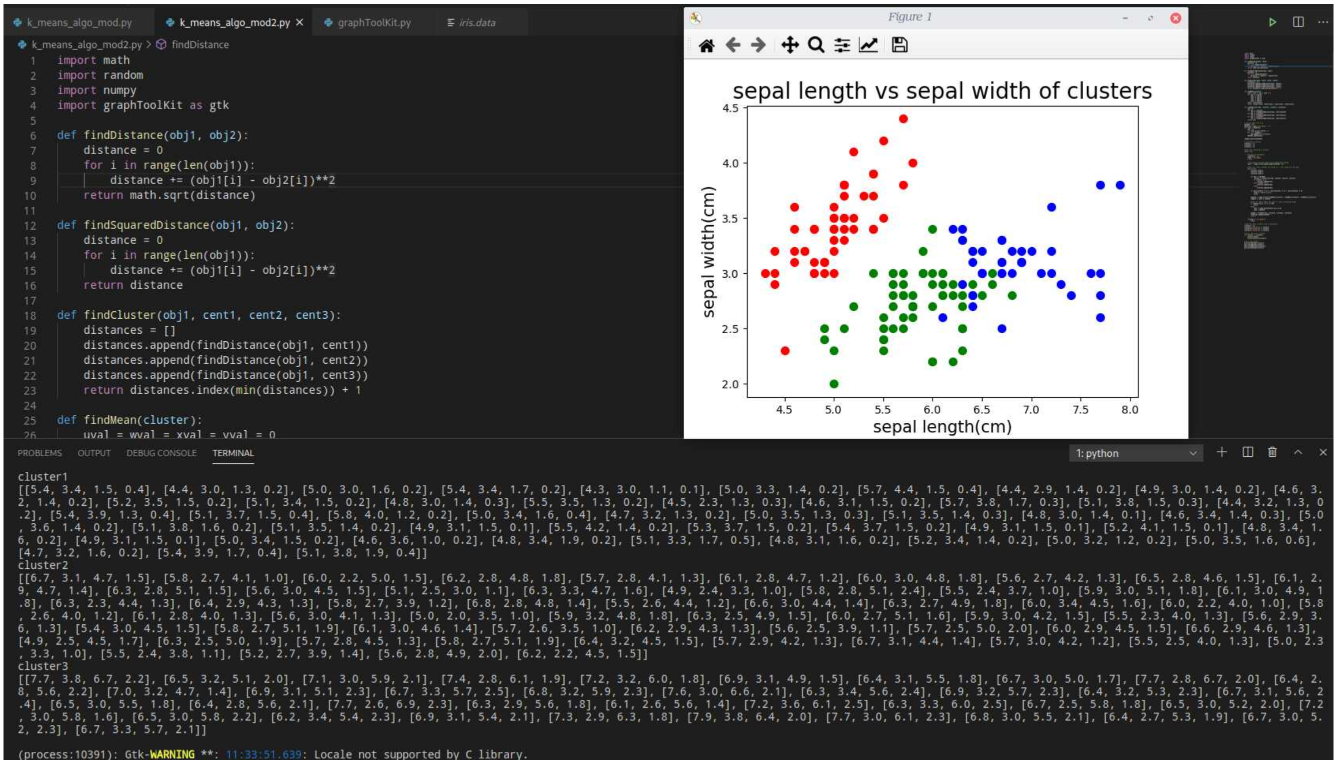
```python
import math
import random
import numpy
import graphToolKit as gtk

def findDistance(obj1, obj2):
    distance = 0
    for i in range(len(obj1)):
        distance += (obj1[i] - obj2[i])**2
    return math.sqrt(distance)

def findSquaredDistance(obj1, obj2):
    distance = 0
    for i in range(len(obj1)):
        distance += (obj1[i] - obj2[i])**2
    return distance

def findCluster(obj1, cent1, cent2, cent3):
    distances = []
    distances.append(findDistance(obj1, cent1))
    distances.append(findDistance(obj1, cent2))
    distances.append(findDistance(obj1, cent3))
    return distances.index(min(distances)) + 1

def findMean(cluster):
    uval = wval = xval = yval = 0
```

```
cluster1
[[5.4, 3.4, 1.5, 0.4], [4.4, 3.0, 1.3, 0.2], [5.0, 3.0, 1.6, 0.2], [5.4, 3.4, 1.7, 0.2], [4.3, 3.0, 1.1, 0.1], [5.0, 3.3, 1.4, 0.2], [5.7, 4.4, 1.5, 0.4], [4.4, 2.9, 1.4, 0.2], [4.9, 3.0, 1.4, 0.2], [4.6, 3.
2, 1.4, 0.2], [5.2, 3.5, 1.5, 0.2], [5.1, 3.4, 1.5, 0.2], [4.8, 3.0, 1.4, 0.3], [5.5, 3.5, 1.3, 0.2], [4.5, 2.3, 1.3, 0.3], [4.6, 3.1, 1.5, 0.2], [5.7, 3.8, 1.7, 0.3], [5.1, 3.8, 1.5, 0.3], [4.4, 3.2, 1.3, 0
.2], [5.4, 3.9, 1.3, 0.4], [5.1, 3.7, 1.5, 0.4], [5.8, 4.0, 1.2, 0.2], [5.0, 3.4, 1.6, 0.4], [4.7, 3.2, 1.3, 0.2], [5.0, 3.5, 1.3, 0.3], [5.1, 3.5, 1.4, 0.3], [4.8, 3.0, 1.4, 0.1], [4.6, 3.4, 1.4, 0.3], [5.0
, 3.6, 1.4, 0.2], [5.1, 3.8, 1.6, 0.2], [5.1, 3.5, 1.4, 0.2], [4.9, 3.1, 1.5, 0.1], [5.5, 4.2, 1.4, 0.2], [5.3, 3.7, 1.5, 0.2], [5.4, 3.7, 1.5, 0.2], [4.9, 3.1, 1.5, 0.1], [5.2, 4.1, 1.5, 0.1], [4.8, 3.4, 1.
6, 0.2], [4.9, 3.1, 1.5, 0.1], [5.0, 3.4, 1.5, 0.2], [4.6, 3.6, 1.0, 0.2], [4.8, 3.4, 1.9, 0.2], [5.1, 3.3, 1.7, 0.5], [4.8, 3.1, 1.6, 0.2], [5.2, 3.4, 1.4, 0.2], [5.0, 3.2, 1.2, 0.2], [5.0, 3.5, 1.6, 0.6],
[4.7, 3.2, 1.6, 0.2], [5.4, 3.9, 1.7, 0.4], [5.1, 3.8, 1.9, 0.4]]
cluster2
[[6.7, 3.1, 4.7, 1.5], [5.8, 2.7, 4.1, 1.0], [6.0, 2.2, 5.0, 1.5], [6.2, 2.8, 4.8, 1.8], [5.7, 2.8, 4.1, 1.3], [6.1, 2.8, 4.7, 1.2], [6.0, 3.0, 4.8, 1.8], [5.6, 2.7, 4.2, 1.3], [6.5, 2.8, 4.6, 1.5], [6.1, 2.
9, 4.7, 1.4], [6.3, 2.8, 5.1, 1.5], [5.6, 3.0, 4.5, 1.5], [5.1, 2.5, 3.0, 1.1], [6.3, 3.3, 4.7, 1.6], [4.9, 2.4, 3.3, 1.0], [5.8, 2.8, 5.1, 2.4], [5.5, 2.4, 3.7, 1.0], [5.9, 3.0, 5.1, 1.8], [6.1, 3.0, 4.9, 1
.8], [6.3, 2.3, 4.4, 1.3], [6.4, 2.9, 4.3, 1.3], [5.8, 2.7, 3.9, 1.2], [6.8, 2.8, 4.8, 1.4], [5.5, 2.6, 4.4, 1.2], [6.6, 3.0, 4.4, 1.4], [6.3, 2.7, 4.9, 1.8], [6.0, 3.4, 4.5, 1.6], [6.0, 2.2, 4.0, 1.0], [5.8
, 2.6, 4.0, 1.2], [6.1, 2.8, 4.0, 1.3], [5.6, 3.0, 4.1, 1.3], [5.0, 2.0, 3.5, 1.0], [5.9, 3.2, 4.8, 1.8], [6.3, 2.5, 4.9, 1.5], [6.0, 2.7, 5.1, 1.6], [5.9, 3.0, 4.2, 1.5], [5.5, 2.3, 4.0, 1.3], [5.6, 2.9, 3.
6, 1.3], [5.4, 3.0, 4.5, 1.5], [5.8, 2.7, 5.1, 1.9], [6.1, 3.0, 4.6, 1.4], [5.7, 2.6, 3.5, 1.0], [6.2, 2.9, 4.3, 1.3], [5.6, 2.5, 3.9, 1.1], [5.7, 2.5, 5.0, 2.0], [6.0, 2.9, 4.5, 1.5], [6.6, 2.9, 4.6, 1.3],
[4.9, 2.5, 4.5, 1.7], [6.3, 2.5, 5.0, 1.9], [5.7, 2.8, 4.5, 1.3], [5.8, 2.7, 5.1, 1.9], [6.4, 3.2, 4.5, 1.5], [5.7, 2.9, 4.2, 1.3], [6.7, 3.1, 4.4, 1.4], [5.7, 3.0, 4.2, 1.2], [5.5, 2.5, 4.0, 1.3], [5.0, 2.3
, 3.3, 1.0], [5.5, 2.4, 3.8, 1.1], [5.2, 2.7, 3.9, 1.4], [5.6, 2.8, 4.9, 2.0], [6.2, 2.2, 4.5, 1.5]]
cluster3
[[7.7, 3.8, 6.7, 2.2], [6.5, 3.2, 5.1, 2.0], [7.1, 3.0, 5.9, 2.1], [7.4, 2.8, 6.1, 1.9], [7.2, 3.2, 6.0, 1.8], [6.9, 3.1, 4.9, 1.5], [6.4, 3.1, 5.5, 1.8], [6.7, 3.0, 5.0, 1.7], [7.7, 2.8, 6.7, 2.0], [6.4, 2.
8, 5.6, 2.2], [7.0, 3.2, 4.7, 1.4], [6.9, 3.1, 5.1, 2.3], [6.7, 3.3, 5.7, 2.5], [6.8, 3.2, 5.9, 2.3], [7.6, 3.0, 6.6, 2.1], [6.3, 3.4, 5.6, 2.4], [6.9, 3.2, 5.7, 2.3], [6.4, 3.2, 5.3, 2.3], [6.7, 3.1, 5.6, 2
.4], [6.5, 3.0, 5.5, 1.8], [6.4, 2.8, 5.6, 2.1], [7.7, 2.6, 6.9, 2.3], [6.3, 2.9, 5.6, 1.8], [6.1, 2.6, 5.6, 1.4], [7.2, 3.6, 6.1, 2.5], [6.3, 3.3, 6.0, 2.5], [6.7, 2.5, 5.8, 1.8], [6.5, 3.0, 5.2, 2.0], [7.2
, 3.0, 5.8, 1.6], [6.5, 3.0, 5.8, 2.2], [6.2, 3.4, 5.4, 2.3], [6.9, 3.1, 5.4, 2.1], [7.3, 2.9, 6.3, 1.8], [7.9, 3.8, 6.4, 2.0], [7.7, 3.0, 6.1, 2.3], [6.8, 3.0, 5.5, 2.1], [6.4, 2.7, 5.3, 1.9], [6.7, 3.0, 5.
2, 2.3], [6.7, 3.3, 5.7, 2.1]]

(process:10391): Gtk-WARNING **: 11:33:51.639: Locale not supported by C library.
        Using the fallback 'C' locale.

(python:10391): Gtk-WARNING **: 11:33:51.670: Theme parsing error: gtk.css:3611:22: 'none' is not a valid color name.
```



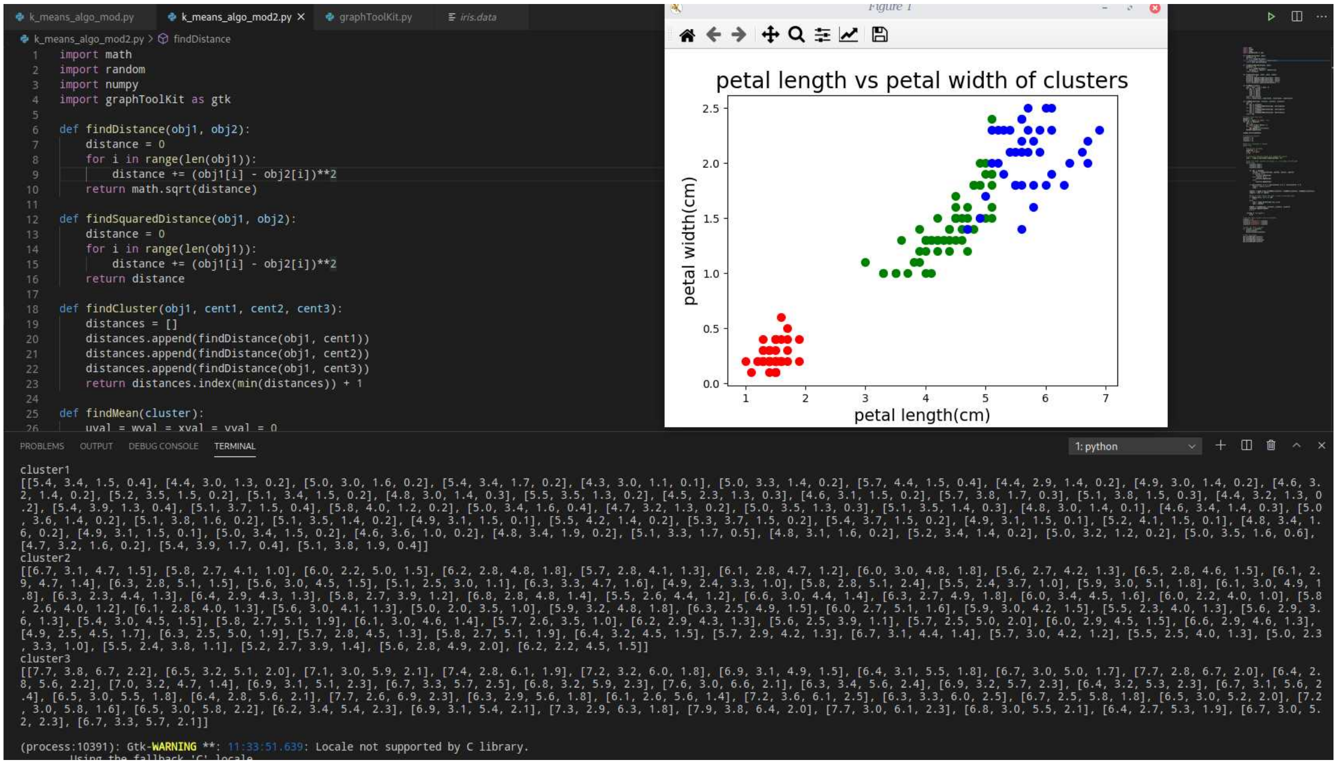
```python
import math
import random
import numpy
import graphToolKit as gtk

def findDistance(obj1, obj2):
    distance = 0
    for i in range(len(obj1)):
        distance += (obj1[i] - obj2[i])**2
    return math.sqrt(distance)

def findSquaredDistance(obj1, obj2):
    distance = 0
    for i in range(len(obj1)):
        distance += (obj1[i] - obj2[i])**2
    return distance

def findCluster(obj1, cent1, cent2, cent3):
    distances = []
    distances.append(findDistance(obj1, cent1))
    distances.append(findDistance(obj1, cent2))
    distances.append(findDistance(obj1, cent3))
    return distances.index(min(distances)) + 1

def findMean(cluster):
    uval = wval = xval = yval = 0
```

```
cluster1
[[5.4, 3.4, 1.5, 0.4], [4.4, 3.0, 1.3, 0.2], [5.0, 3.0, 1.6, 0.2], [5.4, 3.4, 1.7, 0.2], [4.3, 3.0, 1.1, 0.1], [5.0, 3.3, 1.4, 0.2], [5.7, 4.4, 1.5, 0.4], [4.4, 2.9, 1.4, 0.2], [4.9, 3.0, 1.4, 0.2], [4.6, 3.
2, 1.4, 0.2], [5.2, 3.5, 1.5, 0.2], [5.1, 3.4, 1.5, 0.2], [4.8, 3.0, 1.4, 0.3], [5.5, 3.5, 1.3, 0.2], [4.5, 2.3, 1.3, 0.3], [4.6, 3.1, 1.5, 0.2], [5.7, 3.8, 1.7, 0.3], [5.1, 3.8, 1.5, 0.3], [4.4, 3.2, 1.3, 0
.2], [5.4, 3.9, 1.3, 0.4], [5.1, 3.7, 1.5, 0.4], [5.8, 4.0, 1.2, 0.2], [5.0, 3.4, 1.6, 0.4], [4.7, 3.2, 1.3, 0.2], [5.0, 3.5, 1.3, 0.3], [5.1, 3.5, 1.4, 0.3], [4.8, 3.0, 1.4, 0.1], [4.6, 3.4, 1.4, 0.3], [5.0
, 3.6, 1.4, 0.2], [5.1, 3.8, 1.6, 0.2], [5.1, 3.5, 1.4, 0.2], [4.9, 3.1, 1.5, 0.1], [5.5, 4.2, 1.4, 0.2], [5.3, 3.7, 1.5, 0.2], [5.4, 3.7, 1.5, 0.2], [4.9, 3.1, 1.5, 0.1], [5.2, 4.1, 1.5, 0.1], [4.8, 3.4, 1.
6, 0.2], [4.9, 3.1, 1.5, 0.1], [5.0, 3.4, 1.5, 0.2], [4.6, 3.6, 1.0, 0.2], [4.8, 3.4, 1.9, 0.2], [5.1, 3.3, 1.7, 0.5], [4.8, 3.1, 1.6, 0.2], [5.2, 3.4, 1.4, 0.2], [5.0, 3.2, 1.2, 0.2], [5.0, 3.5, 1.6, 0.6],
[4.7, 3.2, 1.6, 0.2], [5.4, 3.9, 1.7, 0.4], [5.1, 3.8, 1.9, 0.4]]
cluster2
[[6.7, 3.1, 4.7, 1.5], [5.8, 2.7, 4.1, 1.0], [6.0, 2.2, 5.0, 1.5], [6.2, 2.8, 4.8, 1.8], [5.7, 2.8, 4.1, 1.3], [6.1, 2.8, 4.7, 1.2], [6.0, 3.0, 4.8, 1.8], [5.6, 2.7, 4.2, 1.3], [6.5, 2.8, 4.6, 1.5], [6.1, 2.
9, 4.7, 1.4], [6.3, 2.8, 5.1, 1.5], [5.6, 3.0, 4.5, 1.5], [5.1, 2.5, 3.0, 1.1], [6.3, 3.3, 4.7, 1.6], [4.9, 2.4, 3.3, 1.0], [5.8, 2.8, 5.1, 2.4], [5.5, 2.4, 3.7, 1.0], [5.9, 3.0, 5.1, 1.8], [6.1, 3.0, 4.9, 1
.8], [6.3, 2.3, 4.4, 1.3], [6.4, 2.9, 4.3, 1.3], [5.8, 2.7, 3.9, 1.2], [6.8, 2.8, 4.8, 1.4], [5.5, 2.6, 4.4, 1.2], [6.6, 3.0, 4.4, 1.4], [6.3, 2.7, 4.9, 1.8], [6.0, 3.4, 4.5, 1.6], [6.0, 2.2, 4.0, 1.0], [5.8
, 2.6, 4.0, 1.2], [6.1, 2.8, 4.0, 1.3], [5.6, 3.0, 4.1, 1.3], [5.0, 2.0, 3.5, 1.0], [5.9, 3.2, 4.8, 1.8], [6.3, 2.5, 4.9, 1.5], [6.0, 2.7, 5.1, 1.6], [5.9, 3.0, 4.2, 1.5], [5.5, 2.3, 4.0, 1.3], [5.6, 2.9, 3.
6, 1.3], [5.4, 3.0, 4.5, 1.5], [5.8, 2.7, 5.1, 1.9], [6.1, 3.0, 4.6, 1.4], [5.7, 2.6, 3.5, 1.0], [6.2, 2.9, 4.3, 1.3], [5.6, 2.5, 3.9, 1.1], [5.7, 2.5, 5.0, 2.0], [6.0, 2.9, 4.5, 1.5], [6.6, 2.9, 4.6, 1.3],
[4.9, 2.5, 4.5, 1.7], [6.3, 2.5, 5.0, 1.9], [5.7, 2.8, 4.5, 1.3], [5.8, 2.7, 5.1, 1.9], [6.4, 3.2, 4.5, 1.5], [5.7, 2.9, 4.2, 1.3], [6.7, 3.1, 4.4, 1.4], [5.7, 3.0, 4.2, 1.2], [5.5, 2.5, 4.0, 1.3], [5.0, 2.3
, 3.3, 1.0], [5.5, 2.4, 3.8, 1.1], [5.2, 2.7, 3.9, 1.4], [5.6, 2.8, 4.9, 2.0], [6.2, 2.2, 4.5, 1.5]]
cluster3
[[7.7, 3.8, 6.7, 2.2], [6.5, 3.2, 5.1, 2.0], [7.1, 3.0, 5.9, 2.1], [7.4, 2.8, 6.1, 1.9], [7.2, 3.2, 6.0, 1.8], [6.9, 3.1, 4.9, 1.5], [6.4, 3.1, 5.5, 1.8], [6.7, 3.0, 5.0, 1.7], [7.7, 2.8, 6.7, 2.0], [6.4, 2.
8, 5.6, 2.2], [7.0, 3.2, 4.7, 1.4], [6.9, 3.1, 5.1, 2.3], [6.7, 3.3, 5.7, 2.5], [6.8, 3.2, 5.9, 2.3], [7.6, 3.0, 6.6, 2.1], [6.3, 3.4, 5.6, 2.4], [6.9, 3.2, 5.7, 2.3], [6.4, 3.2, 5.3, 2.3], [6.7, 3.1, 5.6, 2
.4], [6.5, 3.0, 5.5, 1.8], [6.4, 2.8, 5.6, 2.1], [7.7, 2.6, 6.9, 2.3], [6.3, 2.9, 5.6, 1.8], [6.1, 2.6, 5.6, 1.4], [7.2, 3.6, 6.1, 2.5], [6.3, 3.3, 6.0, 2.5], [6.7, 2.5, 5.8, 1.8], [6.5, 3.0, 5.2, 2.0], [7.2
, 3.0, 5.8, 1.6], [6.5, 3.0, 5.8, 2.2], [6.2, 3.4, 5.4, 2.3], [6.9, 3.1, 5.4, 2.1], [7.3, 2.9, 6.3, 1.8], [7.9, 3.8, 6.4, 2.0], [7.7, 3.0, 6.1, 2.3], [6.8, 3.0, 5.5, 2.1], [6.4, 2.7, 5.3, 1.9], [6.7, 3.0, 5.
2, 2.3], [6.7, 3.3, 5.7, 2.1]]

(process:10391): Gtk-WARNING **: 11:33:51.639: Locale not supported by C library.
        Using the fallback 'C' locale.
```



```python
import math
import random
import numpy
import graphToolKit as gtk

def findDistance(obj1, obj2):
    distance = 0
    for i in range(len(obj1)):
        distance += (obj1[i] - obj2[i])**2
    return math.sqrt(distance)

def findSquaredDistance(obj1, obj2):
    distance = 0
    for i in range(len(obj1)):
        distance += (obj1[i] - obj2[i])**2
    return distance

def findCluster(obj1, cent1, cent2, cent3):
    distances = []
    distances.append(findDistance(obj1, cent1))
    distances.append(findDistance(obj1, cent2))
    distances.append(findDistance(obj1, cent3))
    return distances.index(min(distances)) + 1

def findMean(cluster):
    uval = wval = xval = yval = 0
```

```
cluster1
[[5.4, 3.4, 1.5, 0.4], [4.4, 3.0, 1.3, 0.2], [5.0, 3.0, 1.6, 0.2], [5.4, 3.4, 1.7, 0.2], [4.3, 3.0, 1.1, 0.1], [5.0, 3.3, 1.4, 0.2], [5.7, 4.4, 1.5, 0.4], [4.4, 2.9, 1.4, 0.2], [4.9, 3.0, 1.4, 0.2], [4.6, 3.
2, 1.4, 0.2], [5.2, 3.5, 1.5, 0.2], [5.1, 3.4, 1.5, 0.2], [4.8, 3.0, 1.4, 0.3], [5.5, 3.5, 1.3, 0.2], [4.5, 2.3, 1.3, 0.3], [4.6, 3.1, 1.5, 0.2], [5.7, 3.8, 1.7, 0.3], [5.1, 3.8, 1.5, 0.3], [4.4, 3.2, 1.3, 0
.2], [5.4, 3.9, 1.3, 0.4], [5.1, 3.7, 1.5, 0.4], [5.8, 4.0, 1.2, 0.2], [5.0, 3.4, 1.6, 0.4], [4.7, 3.2, 1.3, 0.2], [5.0, 3.5, 1.3, 0.3], [5.1, 3.5, 1.4, 0.3], [4.8, 3.0, 1.4, 0.1], [4.6, 3.4, 1.4, 0.3], [5.0
, 3.6, 1.4, 0.2], [5.1, 3.8, 1.6, 0.2], [5.1, 3.5, 1.4, 0.2], [4.9, 3.1, 1.5, 0.1], [5.5, 4.2, 1.4, 0.2], [5.3, 3.7, 1.5, 0.2], [5.4, 3.7, 1.5, 0.2], [4.9, 3.1, 1.5, 0.1], [5.2, 4.1, 1.5, 0.1], [4.8, 3.4, 1.
6, 0.2], [4.9, 3.1, 1.5, 0.1], [5.0, 3.4, 1.5, 0.2], [4.6, 3.6, 1.0, 0.2], [4.8, 3.4, 1.9, 0.2], [5.1, 3.3, 1.7, 0.5], [4.8, 3.1, 1.6, 0.2], [5.2, 3.4, 1.4, 0.2], [5.0, 3.2, 1.2, 0.2], [5.0, 3.5, 1.6, 0.6],
[4.7, 3.2, 1.6, 0.2], [5.4, 3.9, 1.7, 0.4], [5.1, 3.8, 1.9, 0.4]]
cluster2
[[6.7, 3.1, 4.7, 1.5], [5.8, 2.7, 4.1, 1.0], [6.0, 2.2, 5.0, 1.5], [6.2, 2.8, 4.8, 1.8], [5.7, 2.8, 4.1, 1.3], [6.1, 2.8, 4.7, 1.2], [6.0, 3.0, 4.8, 1.8], [5.6, 2.7, 4.2, 1.3], [6.5, 2.8, 4.6, 1.5], [6.1, 2.
9, 4.7, 1.4], [6.3, 2.8, 5.1, 1.5], [5.6, 3.0, 4.5, 1.5], [5.1, 2.5, 3.0, 1.1], [6.3, 3.3, 4.7, 1.6], [4.9, 2.4, 3.3, 1.0], [5.8, 2.8, 5.1, 2.4], [5.5, 2.4, 3.7, 1.0], [5.9, 3.0, 5.1, 1.8], [6.1, 3.0, 4.9, 1
.8], [6.3, 2.3, 4.4, 1.3], [6.4, 2.9, 4.3, 1.3], [5.8, 2.7, 3.9, 1.2], [6.8, 2.8, 4.8, 1.4], [5.5, 2.6, 4.4, 1.2], [6.6, 3.0, 4.4, 1.4], [6.3, 2.7, 4.9, 1.8], [6.0, 3.4, 4.5, 1.6], [6.0, 2.2, 4.0, 1.0], [5.8
, 2.6, 4.0, 1.2], [6.1, 2.8, 4.0, 1.3], [5.6, 3.0, 4.1, 1.3], [5.0, 2.0, 3.5, 1.0], [5.9, 3.2, 4.8, 1.8], [6.3, 2.5, 4.9, 1.5], [6.0, 2.7, 5.1, 1.6], [5.9, 3.0, 4.2, 1.5], [5.5, 2.3, 4.0, 1.3], [5.6, 2.9, 3.
6, 1.3], [5.4, 3.0, 4.5, 1.5], [5.8, 2.7, 5.1, 1.9], [6.1, 3.0, 4.6, 1.4], [5.7, 2.6, 3.5, 1.0], [6.2, 2.9, 4.3, 1.3], [5.6, 2.5, 3.9, 1.1], [5.7, 2.5, 5.0, 2.0], [6.0, 2.9, 4.5, 1.5], [6.6, 2.9, 4.6, 1.3],
[4.9, 2.5, 4.5, 1.7], [6.3, 2.5, 5.0, 1.9], [5.7, 2.8, 4.5, 1.3], [5.8, 2.7, 5.1, 1.9], [6.4, 3.2, 4.5, 1.5], [5.7, 2.9, 4.2, 1.3], [6.7, 3.1, 4.4, 1.4], [5.7, 3.0, 4.2, 1.2], [5.5, 2.5, 4.0, 1.3], [5.0, 2.3
, 3.3, 1.0], [5.5, 2.4, 3.8, 1.1], [5.2, 2.7, 3.9, 1.4], [5.6, 2.8, 4.9, 2.0], [6.2, 2.2, 4.5, 1.5]]
cluster3
[[7.7, 3.8, 6.7, 2.2], [6.5, 3.2, 5.1, 2.0], [7.1, 3.0, 5.9, 2.1], [7.4, 2.8, 6.1, 1.9], [7.2, 3.2, 6.0, 1.8], [6.9, 3.1, 4.9, 1.5], [6.4, 3.1, 5.5, 1.8], [6.7, 3.0, 5.0, 1.7], [7.7, 2.8, 6.7, 2.0], [6.4, 2.
8, 5.6, 2.2], [7.0, 3.2, 4.7, 1.4], [6.9, 3.1, 5.1, 2.3], [6.7, 3.3, 5.7, 2.5], [6.8, 3.2, 5.9, 2.3], [7.6, 3.0, 6.6, 2.1], [6.3, 3.4, 5.6, 2.4], [6.9, 3.2, 5.7, 2.3], [6.4, 3.2, 5.3, 2.3], [6.7, 3.1, 5.6, 2
.4], [6.5, 3.0, 5.5, 1.8], [6.4, 2.8, 5.6, 2.1], [7.7, 2.6, 6.9, 2.3], [6.3, 2.9, 5.6, 1.8], [6.1, 2.6, 5.6, 1.4], [7.2, 3.6, 6.1, 2.5], [6.3, 3.3, 6.0, 2.5], [6.7, 2.5, 5.8, 1.8], [6.5, 3.0, 5.2, 2.0], [7.2
, 3.0, 5.8, 1.6], [6.5, 3.0, 5.8, 2.2], [6.2, 3.4, 5.4, 2.3], [6.9, 3.1, 5.4, 2.1], [7.3, 2.9, 6.3, 1.8], [7.9, 3.8, 6.4, 2.0], [7.7, 3.0, 6.1, 2.3], [6.8, 3.0, 5.5, 2.1], [6.4, 2.7, 5.3, 1.9], [6.7, 3.0, 5.
2, 2.3], [6.7, 3.3, 5.7, 2.1]]

(process:10391): Gtk-WARNING **: 11:33:51.639: Locale not supported by C library.
```



```python
import math
import random
import numpy
import graphToolKit as gtk

def findDistance(obj1, obj2):
    distance = 0
    for i in range(len(obj1)):
        distance += (obj1[i] - obj2[i])**2
    return math.sqrt(distance)

def findSquaredDistance(obj1, obj2):
    distance = 0
    for i in range(len(obj1)):
        distance += (obj1[i] - obj2[i])**2
    return distance

def findCluster(obj1, cent1, cent2, cent3):
    distances = []
    distances.append(findDistance(obj1, cent1))
    distances.append(findDistance(obj1, cent2))
    distances.append(findDistance(obj1, cent3))
    return distances.index(min(distances)) + 1

def findMean(cluster):
    uval = wval = xval = yval = 0
```

```
cluster1
[[5.4, 3.4, 1.5, 0.4], [4.4, 3.0, 1.3, 0.2], [5.0, 3.0, 1.6, 0.2], [5.4, 3.4, 1.7, 0.2], [4.3, 3.0, 1.1, 0.1], [5.0, 3.3, 1.4, 0.2], [5.7, 4.4, 1.5, 0.4], [4.4, 2.9, 1.4, 0.2], [4.9, 3.0, 1.4, 0.2], [4.6, 3.
2, 1.4, 0.2], [5.2, 3.5, 1.5, 0.2], [5.1, 3.4, 1.5, 0.2], [4.8, 3.0, 1.4, 0.3], [5.5, 3.5, 1.3, 0.2], [4.5, 2.3, 1.3, 0.3], [4.6, 3.1, 1.5, 0.2], [5.7, 3.8, 1.7, 0.3], [5.1, 3.8, 1.5, 0.3], [4.4, 3.2, 1.3, 0
.2], [5.4, 3.9, 1.3, 0.4], [5.1, 3.7, 1.5, 0.4], [5.8, 4.0, 1.2, 0.2], [5.0, 3.4, 1.6, 0.4], [4.7, 3.2, 1.3, 0.2], [5.0, 3.5, 1.3, 0.3], [5.1, 3.5, 1.4, 0.3], [4.8, 3.0, 1.4, 0.1], [4.6, 3.4, 1.4, 0.3], [5.0
, 3.6, 1.4, 0.2], [5.1, 3.8, 1.6, 0.2], [5.1, 3.5, 1.4, 0.2], [4.9, 3.1, 1.5, 0.1], [5.5, 4.2, 1.4, 0.2], [5.3, 3.7, 1.5, 0.2], [5.4, 3.7, 1.5, 0.2], [4.9, 3.1, 1.5, 0.1], [5.2, 4.1, 1.5, 0.1], [4.8, 3.4, 1.
6, 0.2], [4.9, 3.1, 1.5, 0.1], [5.0, 3.4, 1.5, 0.2], [4.6, 3.6, 1.0, 0.2], [4.8, 3.4, 1.9, 0.2], [5.1, 3.3, 1.7, 0.5], [4.8, 3.1, 1.6, 0.2], [5.2, 3.4, 1.4, 0.2], [5.0, 3.2, 1.2, 0.2], [5.0, 3.5, 1.6, 0.6],
[4.7, 3.2, 1.6, 0.2], [5.4, 3.9, 1.7, 0.4], [5.1, 3.8, 1.9, 0.4]]
cluster2
[[6.7, 3.1, 4.7, 1.5], [5.8, 2.7, 4.1, 1.0], [6.0, 2.2, 5.0, 1.5], [6.2, 2.8, 4.8, 1.8], [5.7, 2.8, 4.1, 1.3], [6.1, 2.8, 4.7, 1.2], [6.0, 3.0, 4.8, 1.8], [5.6, 2.7, 4.2, 1.3], [6.5, 2.8, 4.6, 1.5], [6.1, 2.
9, 4.7, 1.4], [6.3, 2.8, 5.1, 1.5], [5.6, 3.0, 4.5, 1.5], [5.1, 2.5, 3.0, 1.1], [6.3, 3.3, 4.7, 1.6], [4.9, 2.4, 3.3, 1.0], [5.8, 2.8, 5.1, 2.4], [5.5, 2.4, 3.7, 1.0], [5.9, 3.0, 5.1, 1.8], [6.1, 3.0, 4.9, 1
.8], [6.3, 2.3, 4.4, 1.3], [6.4, 2.9, 4.3, 1.3], [5.8, 2.7, 3.9, 1.2], [6.8, 2.8, 4.8, 1.4], [5.5, 2.6, 4.4, 1.2], [6.6, 3.0, 4.4, 1.4], [6.3, 2.7, 4.9, 1.8], [6.0, 3.4, 4.5, 1.6], [6.0, 2.2, 4.0, 1.0], [5.8
, 2.6, 4.0, 1.2], [6.1, 2.8, 4.0, 1.3], [5.6, 3.0, 4.1, 1.3], [5.0, 2.0, 3.5, 1.0], [5.9, 3.2, 4.8, 1.8], [6.3, 2.5, 4.9, 1.5], [6.0, 2.7, 5.1, 1.6], [5.9, 3.0, 4.2, 1.5], [5.5, 2.3, 4.0, 1.3], [5.6, 2.9, 3.
6, 1.3], [5.4, 3.0, 4.5, 1.5], [5.8, 2.7, 5.1, 1.9], [6.1, 3.0, 4.6, 1.4], [5.7, 2.6, 3.5, 1.0], [6.2, 2.9, 4.3, 1.3], [5.6, 2.5, 3.9, 1.1], [5.7, 2.5, 5.0, 2.0], [6.0, 2.9, 4.5, 1.5], [6.6, 2.9, 4.6, 1.3],
[4.9, 2.5, 4.5, 1.7], [6.3, 2.5, 5.0, 1.9], [5.7, 2.8, 4.5, 1.3], [5.8, 2.7, 5.1, 1.9], [6.4, 3.2, 4.5, 1.5], [5.7, 2.9, 4.2, 1.3], [6.7, 3.1, 4.4, 1.4], [5.7, 3.0, 4.2, 1.2], [5.5, 2.5, 4.0, 1.3], [5.0, 2.3
, 3.3, 1.0], [5.5, 2.4, 3.8, 1.1], [5.2, 2.7, 3.9, 1.4], [5.6, 2.8, 4.9, 2.0], [6.2, 2.2, 4.5, 1.5]]
cluster3
[[7.7, 3.8, 6.7, 2.2], [6.5, 3.2, 5.1, 2.0], [7.1, 3.0, 5.9, 2.1], [7.4, 2.8, 6.1, 1.9], [7.2, 3.2, 6.0, 1.8], [6.9, 3.1, 4.9, 1.5], [6.4, 3.1, 5.5, 1.8], [6.7, 3.0, 5.0, 1.7], [7.7, 2.8, 6.7, 2.0], [6.4, 2.
8, 5.6, 2.2], [7.0, 3.2, 4.7, 1.4], [6.9, 3.1, 5.1, 2.3], [6.7, 3.3, 5.7, 2.5], [6.8, 3.2, 5.9, 2.3], [7.6, 3.0, 6.6, 2.1], [6.3, 3.4, 5.6, 2.4], [6.9, 3.2, 5.7, 2.3], [6.4, 3.2, 5.3, 2.3], [6.7, 3.1, 5.6, 2
.4], [6.5, 3.0, 5.5, 1.8], [6.4, 2.8, 5.6, 2.1], [7.7, 2.6, 6.9, 2.3], [6.3, 2.9, 5.6, 1.8], [6.1, 2.6, 5.6, 1.4], [7.2, 3.6, 6.1, 2.5], [6.3, 3.3, 6.0, 2.5], [6.7, 2.5, 5.8, 1.8], [6.5, 3.0, 5.2, 2.0], [7.2
, 3.0, 5.8, 1.6], [6.5, 3.0, 5.8, 2.2], [6.2, 3.4, 5.4, 2.3], [6.9, 3.1, 5.4, 2.1], [7.3, 2.9, 6.3, 1.8], [7.9, 3.8, 6.4, 2.0], [7.7, 3.0, 6.1, 2.3], [6.8, 3.0, 5.5, 2.1], [6.4, 2.7, 5.3, 1.9], [6.7, 3.0, 5.
2, 2.3], [6.7, 3.3, 5.7, 2.1]]

(process:10391): Gtk-WARNING **: 11:33:51.639: Locale not supported by C library.
        Using the fallback 'C' locale.
```