Artificial Intelligence

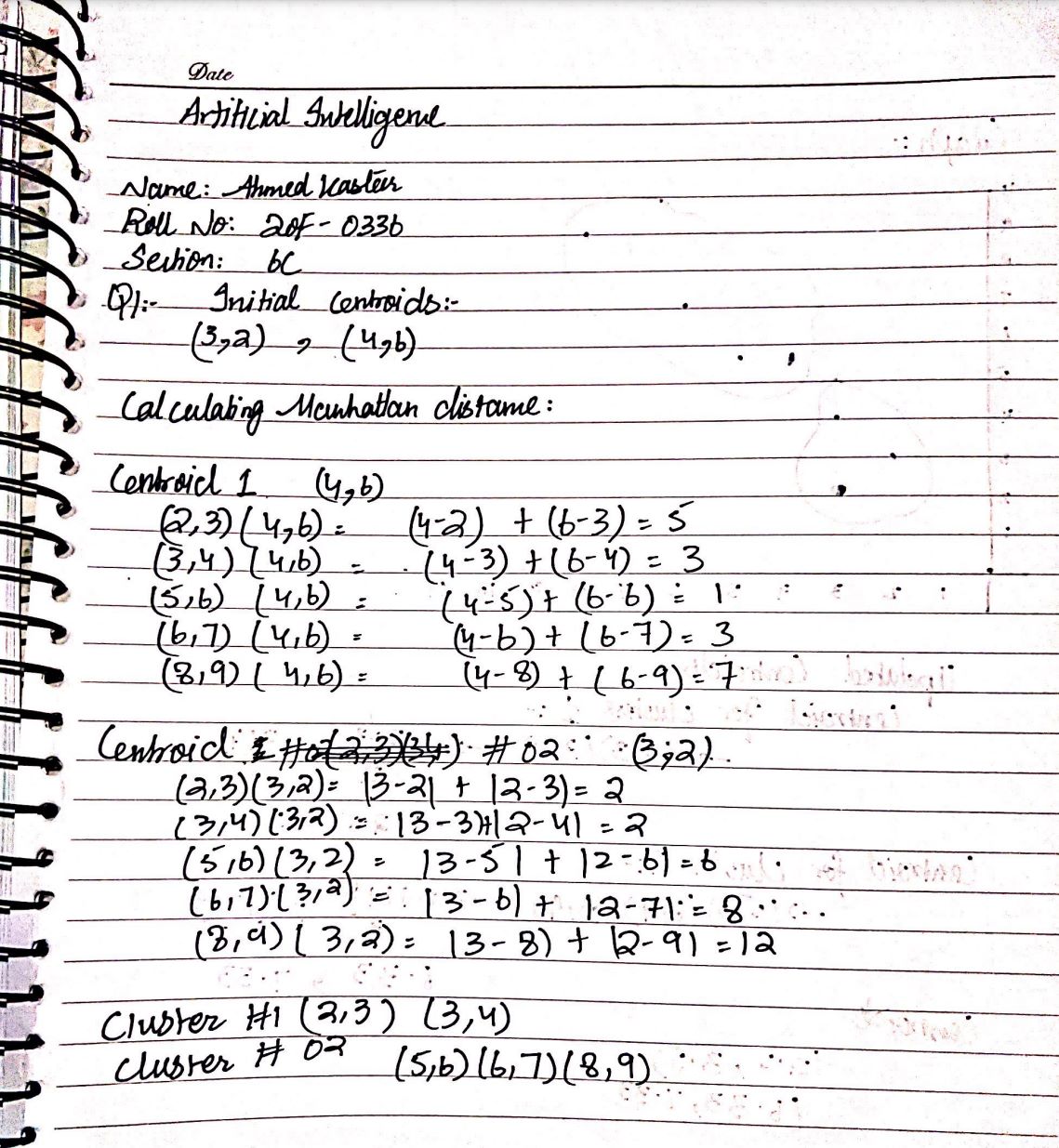
Assignment # 05

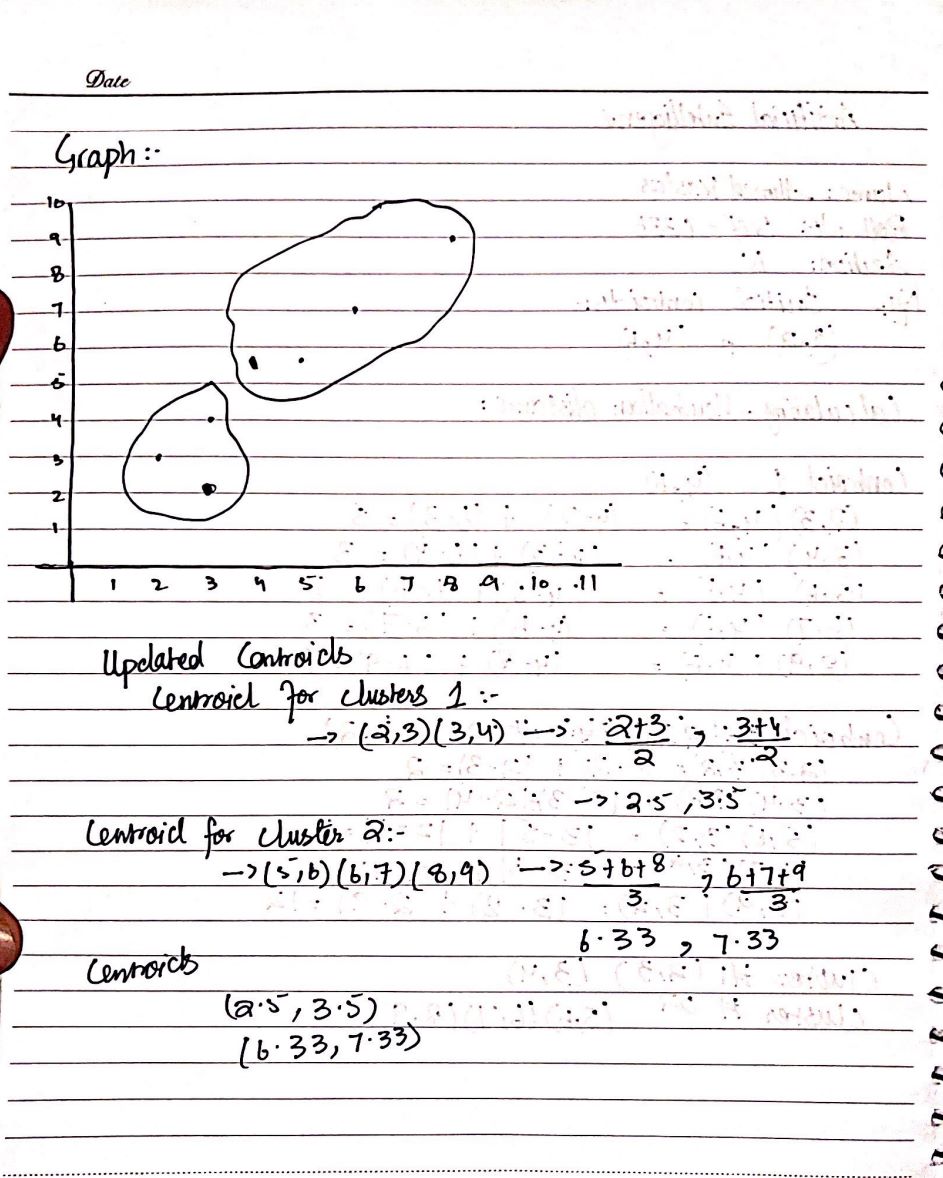
Name: Ahmed Kasteer

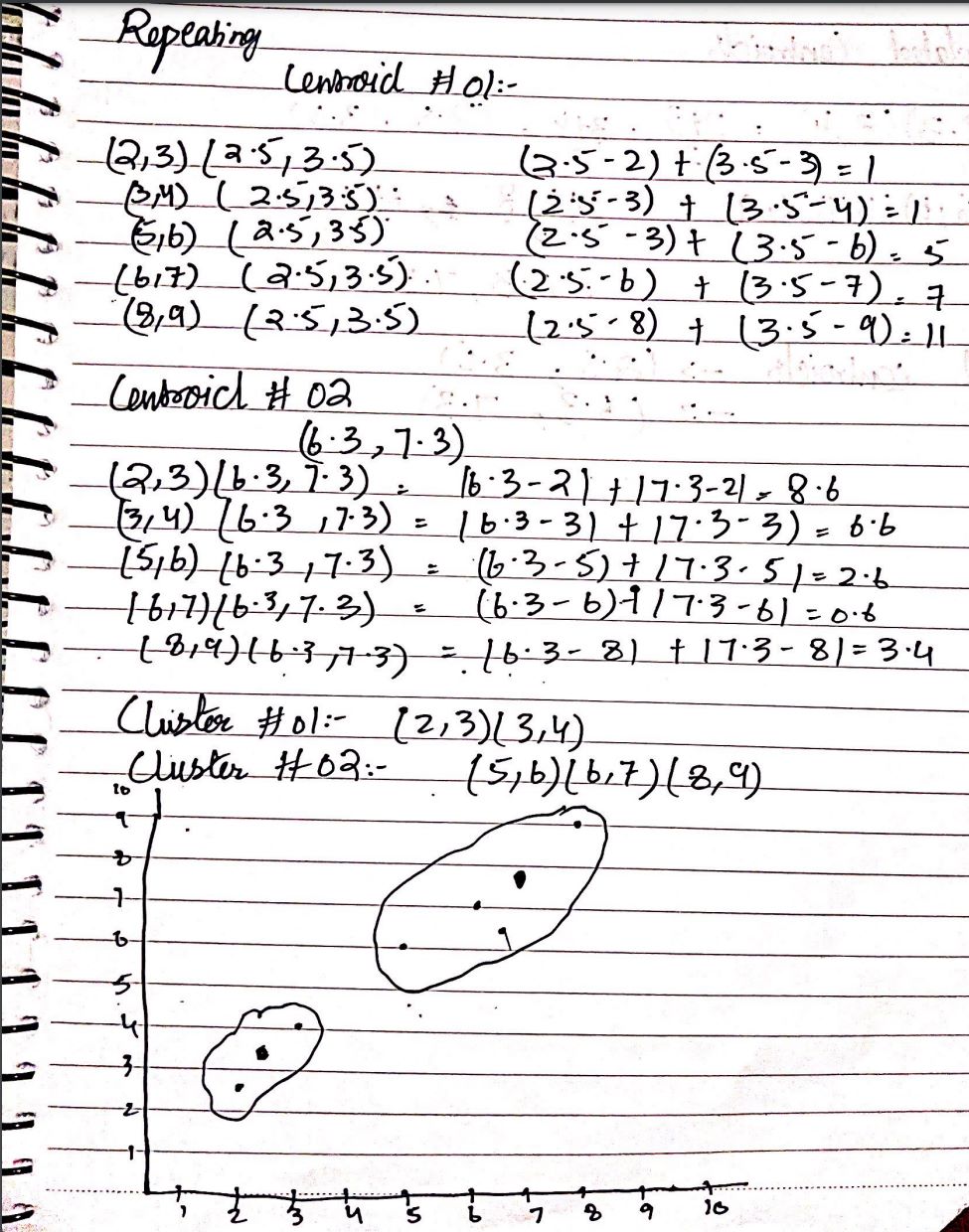
Roll Number: 20F-0336

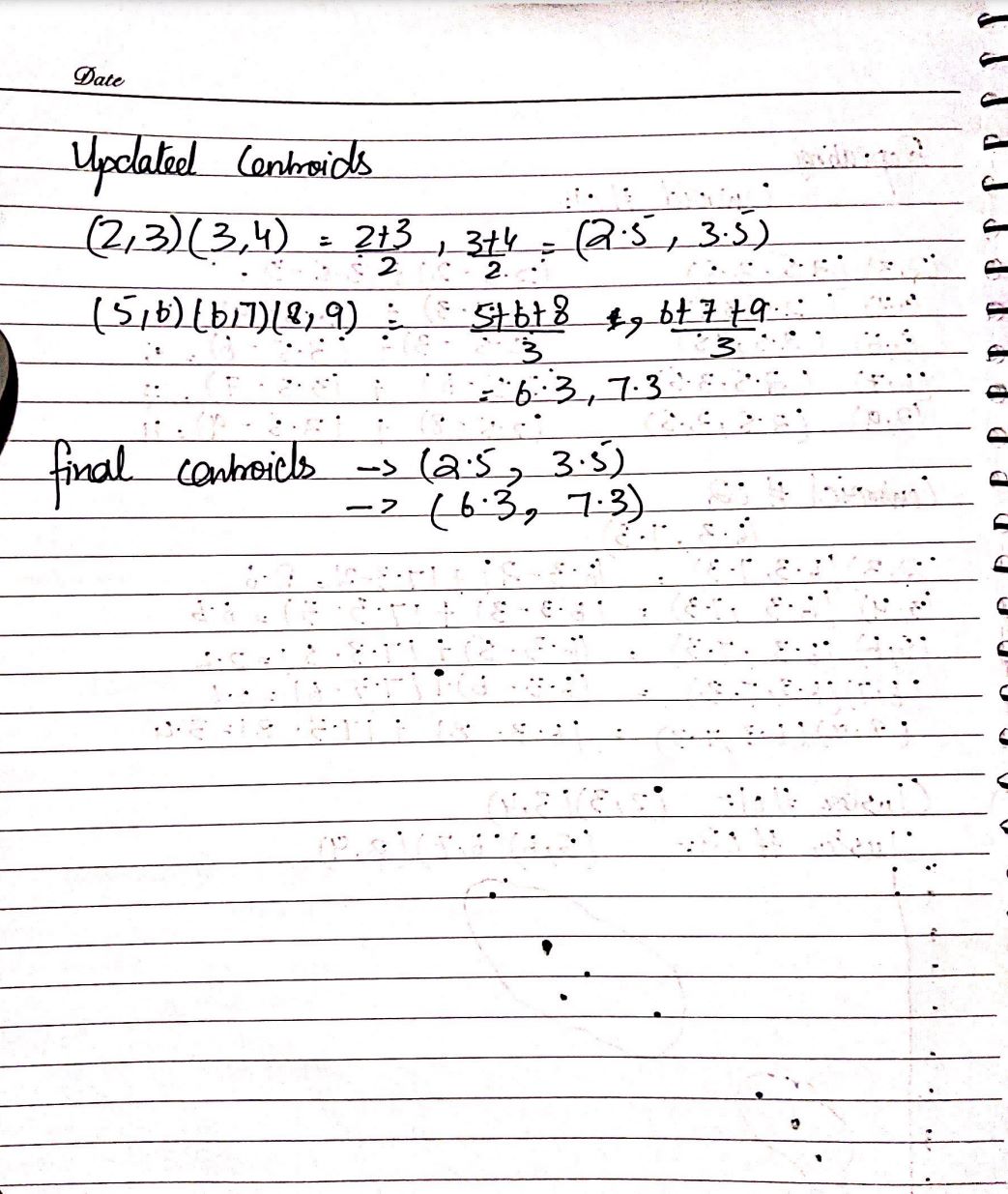
Section: 6C

Question 1:



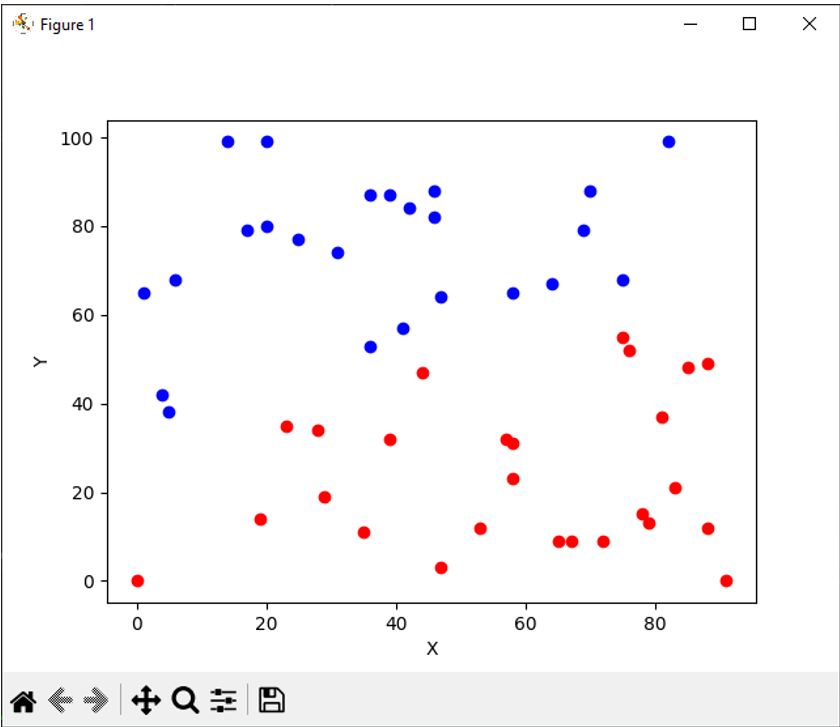






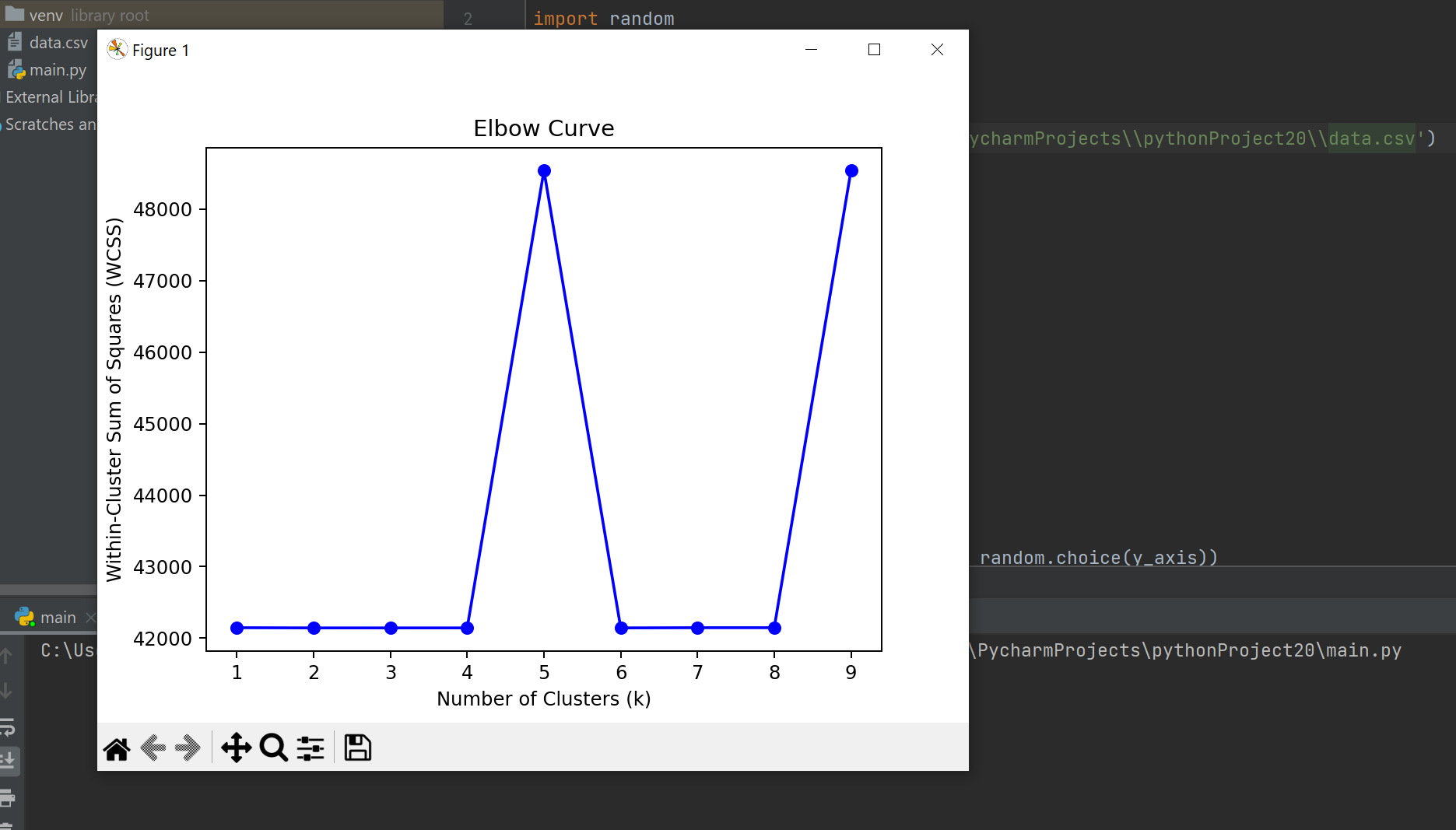
Question No: 2

import pandas as pd  
import random  
import math  
import matplotlib.pyplot as plt  
  
data = pd.read\_csv(r'C:\\Users\\ahmed\\PycharmProjects\\pythonProject20\\data.csv')  
x = pd.DataFrame(data, columns=['X'])  
y = pd.DataFrame(data, columns=['Y'])  
  
x\_axis = []  
y\_axis = []  
for i in range(50):  
 x\_axis.append(x['X'][i])  
 y\_axis.append(y['Y'][i])  
  
centroid\_1 = (random.choice(x\_axis), random.choice(y\_axis))  
centroid\_2 = (random.choice(x\_axis), random.choice(y\_axis))  
  
pre\_1 = None  
pre\_2 = None  
cluster1\_x = []  
cluster1\_y = []  
cluster2\_y = []  
cluster2\_x = []  
  
while pre\_1 != centroid\_1 or pre\_2 != centroid\_2:  
 cluster1\_x.clear()  
 cluster1\_y.clear()  
 cluster2\_x.clear()  
 cluster2\_y.clear()  
  
 for i, j in zip(x\_axis, y\_axis):  
 test = (i, j)  
 if math.dist(test, centroid\_1) < math.dist(test, centroid\_2):  
 cluster1\_x.append(i)  
 cluster1\_y.append(j)  
 else:  
 cluster2\_x.append(i)  
 cluster2\_y.append(j)  
  
 pre\_1 = centroid\_1  
 pre\_2 = centroid\_2  
  
 centroid\_1 = (sum(cluster1\_x) / len(cluster1\_x), sum(cluster1\_y) / len(cluster1\_y))  
 centroid\_2 = (sum(cluster2\_x) / len(cluster2\_x), sum(cluster2\_y) / len(cluster2\_y))  
  
plt.scatter(cluster1\_x, cluster1\_y, c='red', label='Cluster 1')  
plt.scatter(cluster2\_x, cluster2\_y, c='blue', label='Cluster 2')  
plt.xlabel('X')  
plt.ylabel('Y')  
plt.show()



Question 3:

import pandas as pd  
import random  
import math  
import matplotlib.pyplot as plt  
  
data = pd.read\_csv(r'C:\\Users\\ahmed\\PycharmProjects\\pythonProject20\\data.csv')  
x = pd.DataFrame(data, columns=['X'])  
y = pd.DataFrame(data, columns=['Y'])  
  
x\_axis = []  
y\_axis = []  
for i in range(50):  
 x\_axis.append(x['X'][i])  
 y\_axis.append(y['Y'][i])  
  
k\_values = range(1, 10)  
wcss = []  
  
for k in k\_values:  
 centroid\_1 = (random.choice(x\_axis), random.choice(y\_axis))  
 centroid\_2 = (random.choice(x\_axis), random.choice(y\_axis))  
  
 pre\_1 = None  
 pre\_2 = None  
 cluster1\_x = []  
 cluster1\_y = []  
 cluster2\_y = []  
 cluster2\_x = []  
  
 while pre\_1 != centroid\_1 or pre\_2 != centroid\_2:  
 cluster1\_x.clear()  
 cluster1\_y.clear()  
 cluster2\_x.clear()  
 cluster2\_y.clear()  
  
 for i, j in zip(x\_axis, y\_axis):  
 test = (i, j)  
 if math.dist(test, centroid\_1) < math.dist(test, centroid\_2):  
 cluster1\_x.append(i)  
 cluster1\_y.append(j)  
 else:  
 cluster2\_x.append(i)  
 cluster2\_y.append(j)  
  
 pre\_1 = centroid\_1  
 pre\_2 = centroid\_2  
  
 centroid\_1 = (sum(cluster1\_x) / len(cluster1\_x), sum(cluster1\_y) / len(cluster1\_y))  
 centroid\_2 = (sum(cluster2\_x) / len(cluster2\_x), sum(cluster2\_y) / len(cluster2\_y))  
  
 wcss.append(sum(math.dist((i, j), centroid\_1)\*\*2 for i, j in zip(cluster1\_x, cluster1\_y)) +  
 sum(math.dist((i, j), centroid\_2)\*\*2 for i, j in zip(cluster2\_x, cluster2\_y)))  
  
plt.plot(k\_values, wcss, marker='o', linestyle='-', color='b')  
plt.xlabel('Number of Clusters (k)')  
plt.ylabel('Within-Cluster Sum of Squares (WCSS)')  
plt.title('Elbow Curve')  
plt.show()



Question: 4

import pandas as pd  
import random  
import math  
import matplotlib.pyplot as plt  
  
data = pd.read\_csv(r'C:\\Users\\ahmed\\PycharmProjects\\pythonProject20\\data.csv')  
x = pd.DataFrame(data, columns=['X'])  
y = pd.DataFrame(data, columns=['Y'])  
  
x\_axis = []  
y\_axis = []  
for i in range(50):  
 x\_axis.append(x['X'][i])  
 y\_axis.append(y['Y'][i])  
  
centroid\_1 = (random.choice(x\_axis), random.choice(y\_axis))  
centroid\_2 = (random.choice(x\_axis), random.choice(y\_axis))  
centroid\_3 = (random.choice(x\_axis), random.choice(y\_axis))  
  
pre\_1 = None  
pre\_2 = None  
pre\_3 = None  
  
cluster1\_x = []  
cluster1\_y = []  
cluster2\_x = []  
cluster2\_y = []  
cluster3\_x = []  
cluster3\_y = []  
  
while pre\_1 != centroid\_1 or pre\_2 != centroid\_2 or pre\_3 != centroid\_3:  
 cluster1\_x.clear()  
 cluster1\_y.clear()  
 cluster2\_x.clear()  
 cluster2\_y.clear()  
 cluster3\_x.clear()  
 cluster3\_y.clear()  
  
 for i, j in zip(x\_axis, y\_axis):  
 test = (i, j)  
 distances = [  
 math.dist(test, centroid\_1),  
 math.dist(test, centroid\_2),  
 math.dist(test, centroid\_3)  
 ]  
 min\_distance = min(distances)  
  
 if min\_distance == distances[0]:  
 cluster1\_x.append(i)  
 cluster1\_y.append(j)  
 elif min\_distance == distances[1]:  
 cluster2\_x.append(i)  
 cluster2\_y.append(j)  
 else:  
 cluster3\_x.append(i)  
 cluster3\_y.append(j)  
  
 pre\_1 = centroid\_1  
 pre\_2 = centroid\_2  
 pre\_3 = centroid\_3  
  
 centroid\_1 = (sum(cluster1\_x) / len(cluster1\_x), sum(cluster1\_y) / len(cluster1\_y))  
 centroid\_2 = (sum(cluster2\_x) / len(cluster2\_x), sum(cluster2\_y) / len(cluster2\_y))  
 centroid\_3 = (sum(cluster3\_x) / len(cluster3\_x), sum(cluster3\_y) / len(cluster3\_y))  
  
plt.scatter(cluster1\_x, cluster1\_y, c='red', label='Cluster 1')  
plt.scatter(cluster2\_x, cluster2\_y, c='blue', label='Cluster 2')  
plt.scatter(cluster3\_x, cluster3\_y, c='yellow', label='Cluster 3')  
plt.xlabel('X')  
plt.ylabel('Y')  
plt.legend()  
plt.show()

