3.E)Implementation of K-mean algorithm

AIM:

To implement the K-Means algorithm to cluster unlabeled data into K distinct groups based on feature similarity.

CODE:

```
import random

def euclidean_distance(point1, point2):
   squared_diff = [(a - b)**2 for a, b in zip(point1, point2)]
   return sum(squared_diff)**0.5

def k_means(data, k, max_iterations=100):

   centroids = random.sample(data, k)

   for _ in range(max_iterations):

       clusters = [[] for _ in range(k)]
       for point in data:
           distances = [euclidean_distance(point, centroid) for centroid in centroids]
           cluster_index = distances.index(min(distances))
           clusters[cluster_index].append(point)

       new_centroids = []
       for cluster in clusters:
           if cluster:
               new_centroids.append([sum(dim) / len(cluster) for dim in zip(*cluster)])
           else:
               new_centroids.append(centroids[clusters.index(cluster)])

       if new_centroids == centroids:
```

```python
            break

        centroids = new_centroids

    return centroids, clusters

data = [[1, 2], [1.5, 1.8], [5, 8], [8, 8], [1, 0.6], [9, 11]]
k = 2

centroids, clusters = k_means(data, k)

print("Centroids:", centroids)
print("Clusters:", clusters)
```

OUTPUT:

```
Centroids: [[7.333333333333333, 9.0], [1.1666666666666667, 1.4666666666666666]]
Clusters: [[[5, 8], [8, 8], [9, 11]], [[1, 2], [1.5, 1.8], [1, 0.6]]]
```

RESULT:

The code is executed as expected and the output have been verified successfully.