```python

import numpy as np
import matplotlib.pyplot as plt
import random as rd

def distance(x1,x2):
    '''distance euclidienne entre deux points'''
    d=0
    for i in range(len(x1)):
        d+=(x1[i]-x2[i])**2
    return np.sqrt(d)

def centre(liste):
    '''calcule l'isobarycentre de la liste de points'''
    p=len(liste) #le nombre de points
    #p ne pourra pas être nul car il y aura toujours au moins le centre
    n=len(liste[0]) #la dimension de nos points

    c=[] #pour stocker les coordonnées du centre
    for i in range(n):#calcul de la ième coordonnée du centre
        s=0
        for j in range(p):
            s+=liste[j][i]
        c.append(s/p)
    return c

def debut(liste,k):
    '''choisit au hasard k points parmi la liste'''
    #attention rd.choices(liste,k=k) ne focntionne pas car il peut y avoir des répétions
    choix=[]
    while len(choix)<k:
        x=rd.choice(liste)
        if x not in choix:
            choix.append(x)
    return choix

def clusters(Lc,Lx):
    '''Lc est la liste des centres, Lx la liste des points,retourne la liste des clusters'''
    k=len(Lc) #le nombre de clusters
    liste_clusters=[[] for i in range(k)]
    for point in Lx: #on va affecter chaque point à un cluster
        i_min=0
        d_min=+np.inf
        for i in range(k):
            d=distance(point,Lc[i])
            if d<d_min:
                i_min=i
                d_min=d
        liste_clusters[i_min].append(point)
    return liste_clusters

def arret(Lc_old,Lc_new,epsilon):
    k=len(Lc_old)
    test=True
    for i in range(k):
        if distance(Lc_old[i],Lc_new[i])>epsilon:
            test=False
    #le test sera encore True si toutes les distances ont été plus petites que epsilon
    return test



def kmeans(Lx,k):
    Lc=debut(Lx,k) #on initialise aléatoirement les centres
    test=False
    compteur=0 #pour savoir combien d'itérations seront nécessaires
    while test==False:
        compteur+=1
        liste_clusters=clusters(Lc,Lx) #la liste des clusters

```

```
        #on va recalculer tous les centres
        Lc_new=[]
        for liste in liste_clusters:
            Lc_new.append(centre(liste))

        test=arret(Lc,Lc_new,0.01)

        Lc=Lc_new #on actualise les centres

    return Lc,liste_clusters,compteur




def kmeans_tracé(Lx,k):
    numéro=0 #pour numéroter les figures sauvegardées
    #un choix de k couleurs au hasard
    liste_couleurs = [[rd.random(),rd.random(),rd.random()] for i in range(k)]

    Lc=debut(Lx,k) #on initialise aléatoirement les centres

    #tracé de tous les individus et des premiers centres
    liste_x=[coord[0] for coord in Lx]
    liste_y=[coord[1] for coord in Lx]
    plt.plot(liste_x,liste_y,'ob')
    listec_x=[coord[0] for coord in Lc]
    listec_y=[coord[1] for coord in Lc]
    plt.scatter(listec_x,listec_y,s=300,marker='*',color='red')
    plt.savefig('étape'+str(numéro)+'.png')
    plt.show()
    numéro+=1

    test=False
    compteur=0 #pour savoir combien d'itérations seront nécessaires
    while test==False and compteur<3:

        compteur+=1
        liste_clusters=clusters(Lc,Lx) #la liste des clusters

        #le tracé
        i=0
        for liste in liste_clusters:
            couleur=liste_couleurs[i]
            liste_x=[coord[0] for coord in liste]
            liste_y=[coord[1] for coord in liste]
            plt.plot(liste_x,liste_y,'o',color=couleur)
            i+=1
        listec_x=[coord[0] for coord in Lc]
        listec_y=[coord[1] for coord in Lc]
        plt.scatter(listec_x,listec_y,s=300,marker='*',color='red')
        plt.savefig('étape'+str(numéro)+'.png')
        plt.show()

        numéro+=1

        #on va recalculer tous les centres
        Lc_new=[]
        for liste in liste_clusters:
            Lc_new.append(centre(liste))


        test=arret(Lc,Lc_new,0.01)

        Lc=Lc_new #on actualise les centres

        #le tracé
        i=0
        for liste in liste_clusters:
            couleur=liste_couleurs[i]
            liste_x=[coord[0] for coord in liste]
            liste_y=[coord[1] for coord in liste]
```

```
143                 plt.plot(liste_x,liste_y,'o',color=couleur)
144                 i+=1
145             listec_x=[coord[0] for coord in Lc]
146             listec_y=[coord[1] for coord in Lc]
147             plt.scatter(listec_x,listec_y,s=300,marker='*',color='red')
148             plt.savefig('étape'+str(numéro)+'.png')
149             plt.show()
150             numéro+=1
151 
152         print(compteur)
153 
154     ##on récupère un nuage de points pour tester notre programme
155     points_ex=[[24, 6],
156      [31, 27],
157      [47, 18],
158      [11, 1],
159      [13, 0],
160      [47, 30],
161      [45, 37],
162      [47, 38],
163      [38, 33],
164      [41, 39],
165      [22, 46],
166      [8, 32],
167      [55, 36],
168      [13, 33],
169      [53, 0],
170      [51, 20],
171      [4, 38],
172      [3, 41],
173      [43, 32],
174      [34, 49],
175      [49, 37],
176      [46, 19],
177      [50, 37],
178      [43, 27],
179      [15, 40],
180      [11, 36],
181      [29, 25],
182      [57, 35],
183      [26, 6],
184      [58, 29],
185      [33, 7],
186      [29, 7],
187      [59, 24],
188      [1, 2],
189      [17, 15],
190      [46, 21],
191      [48, 2],
192      [6, 37],
193      [18, 12],
194      [43, 41],
195      [36, 35],
196      [6, 5],
197      [32, 43],
198      [12, 17],
199      [4, 46],
200      [50, 28],
201      [19, 19],
202      [57, 22],
203      [26, 19],
204      [12, 24],
205      [94, 19],
206      [69, 13],
207      [92, 20],
208      [67, 1],
209      [100, 17],
210      [74, 1],
211      [65, 20],
212      [81, 16],
213      [72, 6],
214      [78, 16],
```

```
    [89, 22],
    [72, 20],
    [97, 4],
    [82, 2],
    [69, 19],
    [81, 22],
    [67, 5],
    [69, 0],
    [100, 21],
    [82, 14],
    [92, 7],
    [73, 14],
    [69, 3],
    [72, 25],
    [87, 3],
    [81, 24],
    [93, 16],
    [70, 22],
    [82, 21],
    [76, 10],
    [69, 15],
    [89, 16],
    [88, 25],
    [65, 17],
    [79, 22],
    [73, 18],
    [79, 3],
    [69, 3],
    [84, 11],
    [77, 0],
    [69, 23],
    [90, 4],
    [74, 9],
    [72, 2],
    [91, 19],
    [65, 4],
    [74, 1],
    [96, 5],
    [80, 5],
    [92, 20],
    [50, 45],
    [36, 49],
    [58, 75],
    [13, 75],
    [29, 58],
    [31, 20],
    [46, 86],
    [31, 95],
    [56, 2],
    [47, 71],
    [53, 32],
    [30, 32],
    [59, 99],
    [27, 60],
    [59, 88],
    [10, 99],
    [35, 35],
    [34, 72],
    [12, 85],
    [25, 50],
    [52, 69],
    [35, 80],
    [27, 85],
    [53, 47],
    [55, 29],
    [52, 87],
    [12, 85],
    [57, 1],
    [57, 95],
    [46, 94],
    [36, 74],
    [14, 45],
```

```
    [51, 66],
    [46, 15],
    [33, 60],
    [29, 43],
    [29, 34],
    [52, 69],
    [59, 91],
    [11, 94],
    [52, 45],
    [34, 35],
    [27, 70],
    [38, 73],
    [23, 82],
    [24, 45],
    [60, 3],
    [47, 88],
    [10, 36],
    [15, 40]]

kmeans_tracé(points_ex,3)
```