Activity 3

THEME : LISAN - AL - GAIB

—--

Some helpful References:
1. Array creation routines
https://numpy.org/doc/stable/reference/routines.array-creation.html
2. Logic Functions
https://numpy.org/doc/stable/reference/routines.logic.html
3. Broadcasting
https://numpy.org/doc/stable/user/basics.broadcasting.html
4. Linear Algebra
https://numpy.org/doc/stable/reference/routines.linalg.html
5. Random Generation
https://numpy.org/doc/stable/reference/random/legacy.html

—--

The House of Harkonnens, the great spice miners of Arrakis faced a grave dilemma. The extraction of the precious spice, Melange, was fraught with uncertainty. The spice blooms in remote regions, hidden beneath the endless dunes, and its distribution was irregular and unpredictable.

Legend has it that an old wise woman, inspired by the prophecy of the Lisan-al-Gaib, devised a method to unveil the spice's secret locations. She believed that the spice's distribution held patterns, invisible to the naked eye but decipherable through the whispers of data.

Her method, known as the Lisan-al-Gaib Algorithm, became a beacon of hope for the spice miners. This algorithm takes as input the available locations where traces of spice were found(called spice points), and finds out potential locations of spice nodes(or spice centers). Every spice center has a cluster of spice points. Or conversely, every spice point belongs to a cluster of a spice center. The algorithm is explained in the next page.

Fill in the TODOs of the spice.py
When done with all TODOs, to visualise the final results of clustering of spice nodes. Run the run.py script. You can change the data_path and value of K(number of clusters) and observe the output clustering. You are given with **kmeans_1.png** (data_path = 'spice_locations.txt', K = 2) and **kmeans_2.png** (data_path = 'spice_locations2.txt', K = 4)

NOTE:
1. Do not change anything other than where asked for filling the TODOs.
2. You need to achieve the required output without using loops. To check no usage of loops, we shall run a naive approach of parsing of file to search. Each additional for/while => incurs a penalty of -2. So do not use them even within comments!

Following is the Lisan-al-Gaib Algorithm.

—---

Input:

1. Dataset containing 2D coordinates of locations where traces of spice were found,
2. K (guess of number of spice nodes)

Output: Potential locations of K spice centers/nodes

Algorithm:

1. For each of the spice point, maintain a label (0 to K-1, corresponding to the cluster of the spice center it belongs to)
2. Initialise with a random guess for the coordinates of K spice centers, choose them to be randomly K of the spice locations themselves.
3. Loop:
    a. E Step: (Keeping the coordinates of the K spice centers fixed, update the labels)
        i. For each spice point, find the nearest spice center, and assign the spice point to the cluster of that spice center.
        ii. To do this, for each spice point, we first compute distance(here, euclidean) to all the spice centers and assign the label corresponding to that spice node which has the shortest of all distances.\
    b. M Step: (Keeping the labels of spice points fixed, update the coordinates of the spice centers)
        i. The new coordinate of the i^{th} spice center is the mean of coordinates of all spice points belonging to the cluster wiith label = i
    c. Repeat Loop until the labels don't change.
