Description

In this exercise, you are required to implement the k-means algorithm and apply it to a real-life data set.

Input

The provided input file ("places.txt") consists of the locations of 300 places in the US. Each location is a two-dimensional point that represents the longitude and latitude of the place. For example, "-112.1,33.5" means the longitude of the place is -112.1, and the latitude is 33.5.

https://drive.google.com/file/d/1ASPxnDzmgWWybwrvuIqOInyx4T8vkJym/view?usp=sharing

Output

You are required to implement the k-means algorithm and use it to cluster the 300 locations into **three** clusters, such that the locations in the same cluster are geographically close to each other.

After reading in the 300 locations in "places.txt" and applying the k-means algorithm (with k = 3), you are required to generate an output file named "clusters.txt". The output file should contain exactly 300 lines, where each line represents the cluster label of each location. Every line should be in the format: location_id cluster_label.

An example snippet of the output "clusters.txt" file is provided below:

0 1

1 0

2 1

3 2

4 0

In the above, the five lines denote the cluster ids of the first five locations in the input file, which means:

The first location belongs to cluster "1"

The second location belongs to cluster "0"

The third location belongs to cluster "1"

The fourth location belongs to cluster "2"

The fifth location belongs to cluster "0"

Important Tips

When implementing the K-means algorithm, you could use any programming language you like. We only need the generated cluster label file.

Make sure that you format each line correctly in the output file. For instance, use space instead of comma to separate the data point id and the cluster label.

In the input file "places.txt", the id of the location starts from 0. That is, the first line in "places.txt" has id 0, the second line has id 1, ..., and the last place has id 299.

When generating the output file, please note that the order of the cluster labels does not matter. For example, if there are three clusters, you can use either [0, 1, 2] or [2, 1, 0] as labels for them --- it is correct as long as you use three distinct integer ids. Thus, the following two cases will be considered equivalent by the grader:

Case 1:

0 0

1 1

2 2

Case 2:

0 2

1 1

2 0