**Farmer’s Market**

You have access to a farmer’s market data which gives insights into thousands of farmers markets in the U.S. along with details of the produce they support.

The dataset is formulated as a CSV having numerous columns. The data includes:

1. Unique identifier of the market
2. Market name
3. X and Y locations - These denote the longitude and latitude of the market’s centroid
4. Address (except for street name)
5. Various services and offerings from the markets as boolean values Y/N

We want to provide a customer with a way to find the markets (s)he is interested in based on the user’s current location and a few filters.

**Part – I** The program will take three input parameters:

1. Location array (point with latitude and longitude)
2. Radius in meters
3. Array of flags (Credit, Organic, Baked goods, cheese, etc). This array need not be an array of strings. It can be indexes based on how you deem fit

The program should list all available farmer’s markets within the given radius from the input location which provide the services/goods supplied as the array in the third input parameter.

The program’s output will be a two-dimensional array of

1. Farmer market ids
2. Distance in meters from source coordinate provided as input

Sample output: [[FMID1, Distance1], [FMID2, Distance 2], …].

For example, if the input is:

[41.098698, -81.553272], 5000, [Credit, WIC, Bakedgoods, Cheese, Vegetables]

The output can be:

[[1019530, 0], [1019531, 3320]]

**Part -II** In the event the user’s query doesn’t yield any results, the system should provide next-best alternative results that would be able to satisfy her/her filters most optimally in the search radius.

The program’s output will be a two-dimensional array of

1. Farmer market ids
2. Distance in meters from source coordinate provided as input
3. % criteria met by the search result item as a floating point number with precision of 2 decimal places

For example, if the search criteria provided by user is as follows:

[41.098698, -81.553272], 5000, [Credit, WIC, Bakedgoods, Cheese, Crafts, Vegetables]

The output should be

[[1019530, 0, 83.33], [1019531, 3320, 83.33], [1019533, 3352, 83.33], [1005075, 3451, 83.33], [1019545, 4513, 83.33]]

Here, the first two markets are supplying Cheese but not Crafts and the last three markets are supplying Crafts but not Cheese.

**Important**: Note that since the distances are fairly large, we want to take into consideration earth’s curvature while calculating the distances.

For the solution, please explain the following.

* Chosen approach
* The primary data structure employed and the reasons for choosing it
* Space/time complexity of building the data structures as well the search algorithm
* Scope of improvement, if any.