**Point 1: Done**

1. Get (latitude and longitude) of 12 places => A and your home => B. **Done**
2. Spread out set A. Make sure set A points are not too close or too far from each other. **Done**
3. Gather data by roaming around the campus and area=> need to take selfie for each location. **Done**
4. Once at the point, go to the mobile app and get the lat, long from the app. Note it down. **Done**
5. Make a table of the locations + names, 13 entries in total. **Done**

**Point 2: Done**

1. Create a KML file(.kml format, which is XML) out of them using a text editor. **Done**
2. Specifically, each location will be a ‘placemark’ in your .kml file, with a label and co-ordinates. **Done**

<https://developers.google.com/kml/documentation/kml_tut#placemarks>

1. The .kml file with 13 locations will be the starter file for doing visualization and queries. Example: <https://bytes.usc.edu/cs585/f21_d--a--ta/hw/HW3/data/starter_kml.xml> **Done**
2. **NOTE:** Keep your label names 15 character or less including spaces. **Done**
3. Read last para **Done**

**Point 3: Done**

1. Install Google Earth on laptop. **Done**
2. Load .kml file into it => shows your sampled locations. **Done**
3. Take a snapshot of this for submission. **Done**

**Point 4: Done**

1. Install PostGIS on laptop. **Done**
2. Browse docs for the special functions and search for basic tutorials. <https://www.postgresqltutorial.com/postgresql-getting-started/>
3. <https://bytes.usc.edu/cs585/f21_d--a--ta/hw/HW3/clips/queries.mp4> Follow this link to create a DB, create a table, insert data and query it. **Done**

**Point 5: Done**

Query 1: **Done**

1. Compute the convex hull of your 13 points. <http://postgis.net/docs/ST_ConvexHull.html> and <https://stackoverflow.com/questions/10461179/k-nearest-neighbor-query-in-postgis> **Done**
2. Use the result polygon’s coords to create a polygon in your .kml file (add relevant xml to specify the KML polygon’s coords). **Done**
3. Load the .kml file in google earth, verify all your points are inside the polygon => take a screenshot. **Done**
4. NOTE: Even your data points need to have a concave perimeter, the convex hull by definition. <https://www.quora.com/What-are-the-real-life-applications-of-convex-hulls> => about convex-hulls. **Done**
5. Be sure to specify the polygon cords in the long,lat format (no spaces after comma). **Done**

Query 2: **Done**

1. Compute 4 nearest neighbors from your house. **Done**
2. Use the query result to create 4 line segments in your .kml file. **Done**
3. Line(home, n1), Line(home, n2), Line(home, n3), Line(home, n4) **Done**
4. Verify this looks correct using google earth, take a snapshot. **Done**

**Point 6: Done**

1. Use OpenLayers (Javascript library) to visualize your location data. **Done**
2. Store the 13 sampled points, via your web browser in a browser cache area in your local machine, where the data would persists (even after you close down the browser). **Done**
3. You’d read back the stored values and visualize them, using the OpenLayers API. **Done**
4. Use HTML5 localstorage to store and load points. **Done**
5. API calls are in Javascript which you would run via a HTML page. **Done**
6. Test it on Chrome. **Done**
7. Starter code for HTML=> <https://bytes.usc.edu/cs585/f21_d--a--ta/hw/HW3/OL/OL.html> **Done**

**Point 7: Done**

1. Use Tommy Trojan as the center, compute a set of lat-long (i.e spatial) cords that lie along a Spirograph Curve. **Done** <https://www.google.com/search?q=Spirograph+curve&num=100&source=lnms&tbm=isch>
2. Get the cords for Tommy Trojan from google. **Done**
3. Create a new .kml file with Spirograph curve points. **Done**
4. Convert the KML to an ESRI ‘shapefile’, visualize the shapefile data using ArcGIS Online. **Done**
5. TO convert from KML to shapefile => <https://mygeodata.cloud/converter/kml-to-shp> => result is a zip file. **Done**
6. About shapefile and its components: <https://desktop.arcgis.com/en/arcmap/10.3/manage-data/shapefiles/what-is-a-shapefile.htm> and <https://desktop.arcgis.com/en/arcmap/10.3/manage-data/shapefiles/shapefile-file-extensions.htm> **Done**
7. Once you have your shape file, upload it to ArcGIS’ online map creator to view your Spirograph curve-shaped points. Log into ArcGIS (create a public free account at <https://www.arcgis.com/index.html> ) , see HW description for the website maneuver. **Website processing error-> resolved**
8. For Spinograph points creation use the following parametric equations, with R=8, r=1, a=4. **Done**

x(t) = (R+r)\*cos((r/R)\*t) - a\*cos((1+r/R)\*t)

y(t) = (R+r)\*sin((r/R)\*t) - a\*sin((1+r/R)\*t)

1. Use above equations, loop through t from 0.00 to n\*Pi and t is in radians not degree, in steps of 0.01. **Done**
2. This will give you a sequence of (x,y) points that make up the spiro curve. **Done**
3. IT MUST HAVE 8 LOOPS. **Done**
4. Center the spirograph curve on TT by adding the lat, long of the centering location. Use these new Spiro based spatial cords in .kml file. **Done**
5. Use any coding language to generate the cords for spiro curve. **Done**

What all to SUBMIT?

Here is what you need to **submit** (**as a single .zip file**):

\* 13 selfies, from step 1 above [if you don't submit these, you will LOSE 2 points!]

\* your .kml file from step 5 above - with the placemarks, convex hull and nearest-neighbor line segments (**1 point**)

\* a text file (.txt or .sql) with your two queries from step 5 - table creation commands (if you use Postgres and directly specify points in your queries, you won't have table creation commands, in which case you wouldn't need to worry about this part), and the queries themselves (**2 points**)

\* screengrabs from steps 3,5 (**1 point**)

\* a .html file (with the OpenLayers code) from step 6, or a CodePen/jsfiddle link (**1 point**)

\* your Spirograph point generation code, the resulting .kml file ("spiro.kml"), shapefile (this needs to be a .zip) and a screenshot (**1 point**)