CSC 416 – Homework 4 (100 points)

Due Date: Monday, October 31st

In this homework you will develop an app to view Cities that have bike shares like Cincinnati's Red Bike (smaller version of this is the NKU bike share). The user will be able to sort this information by city name or cities distance hard-coded location. In this assignment you will learn how to

- Use JSON data
- Using open source library (SwiftyJSON)
- Compute distance using the Haversine formula

Steps to follow for setup :

1. Clone the project – https://classroom.github.com/assignment-invitations/9735d1e322e7392c7226fbd48a4b9f45
2. When opening this project make sure to open .xcworkspace!

User Interface

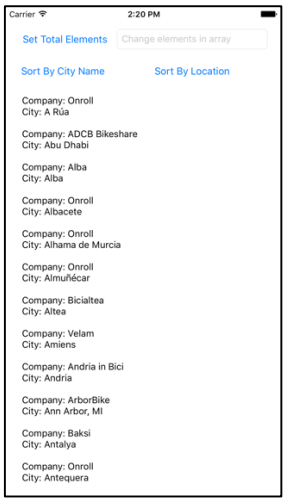
- Textfield
  - get new size of array
  - allow only numbers to be inputted
- TextView
  - show the data
  - uneditable but scrollable
- 3 buttons –

  First Button
  - change size of array

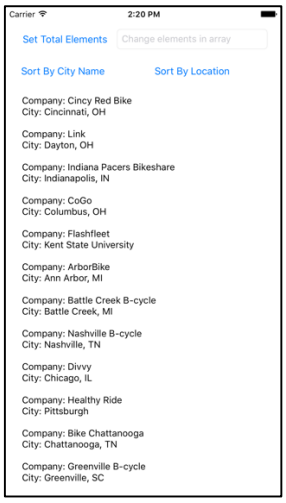
  Second Button
  - sort by city name

  Third Button
  - sort by closest cities

Launch Screen
-Sorted by city names



Sort By Location



Change Total Elements

Implementation Details:

- A TextView will be populated with the company name and city name that is gathered from JSON data.
- The array of cities will be sorted by city name (make sure to sort in a closure) and shown at launch.
- The user will be able to change the size of the array if they want to by specifying size (it could be smaller or bigger) in a TextField and pressing a button.
- When the button is pressed the new sized array will be presented in the TextView.
- The user can sort by city names in alphabetical order. (make sure to sort in a closure)
- The user can also sort by closest city to them (make sure to sort in a closure), we decided to let you hard code the location for the user. A hard coded user location will be provided but you can change the locations if you want. When finished sorting populate the TextView again as well.
- JSON data input file: https://api.citybik.es/v2/networks/
- You are given in the project a library called SwiftyJSON, this will allow you to be able to call the JSON data and parse the returning results. The results are returned back as a Dictionary and so each data has key values you will need to call when parsing it. Save each city into a city struct that has city name, latitude, longitude, company name, and an optional distance to.
  SwiftyJSON Tutorial
  https://www.hackingwithswift.com/read/7/3/parsing-json-data-and-swiftyjson
- How to compute distances:

Haversine Formula
https://rosettacode.org/wiki/Haversine_formula -> shows other languages this was written in
This function finds the distance between 2 locations on a curved surface and returns the distance.

```
func haversine(lat1:Double, lon1:Double, lat2:Double, lon2:Double) -> Double {
        let lat1rad = lat1 * M_PI/180
        let lon1rad = lon1 * M_PI/180
        let lat2rad = lat2 * M_PI/180
        let lon2rad = lon2 * M_PI/180

        let dLat = lat2rad - lat1rad
        let dLon = lon2rad - lon1rad
        var a = sin(dLat/2) * sin(dLat/2)
        a = a + sin(dLon/2) * sin(dLon/2) * cos(lat1rad) * cos(lat2rad)
        let c = 2 * asin(sqrt(a))
        let R = 6372.8
        return R * c
    }
```

What to submit: Push the solution to this problem to GitHub. Once that is submitted then please submit your GitHub user name via Assignment 4 in Blackboard.