LAB TWO

Introduction to Python scripting for Spatial Analysis and Modeling
File Handling & Code Abstraction

Write three Python programs following the instructions below. You may use the given scripts as the templates for your coding. These scripts also include detailed instructions.
Again, a possible approach to accomplish the task is to start from scratch by testing the calculations one at a time rather than trying to modify the whole template at once. In the templates, I highlighted the lines where your code should be put.

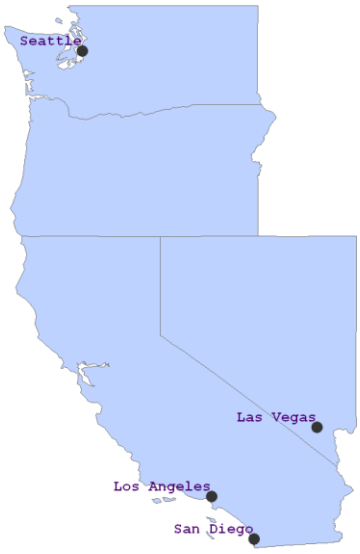
Part One (Script 1)

The following task covers the concept of building modules, function definition, and function invocation.

In part one you will extend the **grid.py** script presented in class. In this script, you will put all the grid functionality from grid.py script from Lab 1 into function definitions. Therefore, you will develop a module of functions that return basic information about 2 dimensional arrays (or grids). The grid.py in the scripts folder is a template for the grid.py module you are developing.
Note that there is a 'testing' section below the function definition section. This is where you should test the functions you define.

Part Two (Script 2 & 3)

The following task covers the concept of file management, class definition, and class instantiation.



city	x_coord	y_coord
Seattle	3	26
Las_Vegas	11	10
Los_Angeles	4	8
San_Diego	7	5

You are going to build a simple application that reads a flat attribute table of cities (above) into a list of city objects and then calculates and displays the distance between these cities.

First you will build a module script **city.py** that defines a **City** class. The class will be an extended version of point_6.py presented in Lecture 3 (Python and OOP). The class will have the following:

Class: City
City Attributes:

- name
- position (x & y)

City Methods:

- Initialization (constructor)
- Description (string that provides basic information about the city – its name and position)
- Calculating distance (between given city and another city)

Note, that the template has an invocation section which can be used for rapid testing of the class you are building. When you are done (i.e. your City objects with their attributes and methods are properly created), **delete this testing section from your city.py**. From now on, you will use the script as a module that can be imported in another application (below).
Next you will complete the given **run_cities.py** application that reads the flat table (cities.txt), builds and prints the city objects, and calculates and prints the distances between these objects (see the sample run below).

As the proof of your work, send me your completed (and tested) scripts via email. **Enjoy** ☺