CIS 1250 PYTHON PROGRAM5 – PROGRAM WITH FUNCTIONS

Turn in Requirements:

5 pts. Name your project LastnameP5, such as GarnerP5.

Program Requirements:

1. 5 pts. Write the file name, your name, email address and purpose of the program at the top of your source code in a comment.

```
# GarnerP5
# Programmer: Rob Garner
# EMail: Rgarner7 @cnm.edu
# Purpose: demonstrate use of functions
```

2. 5 pts. Add comments as appropriate. Be sure that your program output is neatly presented to the user. Add documentation comments to your functions.
3. You are going to write a program that calculates the distance between two geographic points.
4. Your program will have the following functions
   - 4.1. A “header” function that takes no parameters and returns nothing that displays a header. The header will print a summary explaining the purpose of the program.
   - 4.2. A “get_location” function that takes no parameters, asks the user for a latitude and longitude and returns a tuple or list with the latitude and longitude. Make sure you tell the user what units to enter their information in!
   - 4.3. A “distance” function that takes two tuples, each with a latitude and longitude, calculates the distance between those two geographic points and returns the distance.
   - 4.4. The haversine formula is:

Given:

Latitude in decimal degrees: Lat1 and Lat2
Longitude in decimal degrees:Lon1 and Lon2
Radius of earth R (mean radius = 6,371km);

The distance D between points (Lat1, Lon2) and (Lat2, Lon2) can be calculated using:

$$A = \sin^2((Lat1-Lat2)/2) + \cos Lat1 \cdot \cos Lat2 \cdot \sin^2((Lon1-Lon2)/2)$$

$$C = 2 \cdot \operatorname{atan2}(\sqrt{A}, \sqrt{(1-A)})$$

$$D = R \cdot C$$

5. In the main part of the program (after declaring your functions):
   5.1. Call the Header function that displays a header.
   5.2. Program should allow the user to do multiple calculations. Use a “do another?” loop.
   5.3. Inside the loop the program will do the following:
      5.3.1. Call the get_location function to get the first location.
      5.3.2. Call the get_location function again to get the second location.
      5.3.3. Call the distance function passing in the two locations above as arguments.
      5.3.4. Display a nicely formatted message to the user telling them the distance between those two locations.
      5.3.5. Finally ask the user if they want to do another.
6. When done display a good bye message outside the loop.

Testing:

Test your program by finding the Lat and Long for Albuquerque and the Lat and Long for Santa Fe using google maps.

Hints:

See https://www.movable-type.co.uk/scripts/latlong.html for more information on the formula and a solution in JavaScript.

Latitude and longitude can be expressed in Degrees, minutes and seconds or decimal degrees. If you tell the user to input the Latitude and Longitude as decimal degrees, it will be much easier to do this math so you don't have to convert from hours minutes and seconds.

The function that gets the point coordinates from the user will be called twice. Once for each point. Your code will look something like:

```
point1 = GetLocation()

point2 = GetLocation()
```

GetLocation() will return a tuple. Something like (lat, long) depending on what variable names you use in your function.

If you use raw_input to get the coordinates don't forget to convert the string into a number (something like float(x)).

The function that returns distance will be called this way:

```
distance = Distance (point1, point2)
```

point1 and point2 will be tuples that you got out of GetCoords() function.