HW 3 - Distance between two GPS Coordinates (20 points)

Due by Friday February 25th at the start of class at 9:20AM.

In this homework assignment you are going to build on the zip code program we started in class.

Your program should take two zip codes as command line arguments and print the distance, in miles, between the two as well as other information nicely formatted. For example, if the executable was called **`zipdist`** the output might look something like …

```
zipdist 50227 80757
```

```
The distance from POPEJOY,IA to LAST CHANCE,CO is 557.6 miles.
```

Go back to hardcoding the path to the zipcode file in the C program and not using it as a command line argument.

One formula for calculating the *great circle distance* between two latitude and longitude coordinates (coordinates on a sphere) is called the *haversine formula* ...

Let ***(lng1, lat1)*** and ***(lng2, lat2)*** be the coordinates of two locations that we are calculating the distance between.

Let ***R*** be the radius of the earth as 6,371,000 meters.

$$x = sin^2((lat_2 - lat_1)/2) + cos(lat_1)cos(lat_2)sin^2((lng_2 - lng_1)/2)$$

$$y = 2atan2(\sqrt{x}, \sqrt{1 - x})$$

$$distance = Ry$$

Also, in C, the trigonometric function **`sin`**, **`cos`**, **`atan2`** (**`#include math.h`**) take arguments in *radians*. To convert degrees to radians multiply by $\pi/180$.

What to turn in

Create a directory **hw3** and create files **distance.c**, **distance.h**, and **main.c**. The **distance.c** file should have the function that takes two GPS coordinates and returns the distance between the two. Push the files to your course repo by the due date.

Think about the structure of your program and use functions appropriately. Don't stick everything in **main**.

Good programs never crash, they fail gracefully with appropriate error messages.

- What if there are too few command line arguments, or too many?
- What if one of the arguments is a zip code that does not exist?