CENG 112 – DATA STRUCTURES
Homework 2b

March 17, 2017

Due Date: March 27, 2017

Assignment 1 GPS Paths

Write a class named `GPSPath` that keeps a set of `GPSCoord` objects in a private field named `m_points` which is of type `std::vector<GPSCoord>`. Add public functions to `GPSPath` named

- `add_point` that adds a new GPS coordinate to the `GPSPath` from the longtitude and lattitude given as parameters
- `total_distance` that computes and returns the total distance between each pair of consecutive GPS coordinates stored in `m_points`.
- `print` that prints the lattitude and longtitude of each point in the path as

  `(<lat0>,<lon0>) -- (<lat1>,<lon1>) -- (<lat2>,<lon2>) -- (<lat3>,<lon3>)`

  where `<lat_i>` and `<lon_i>` are the lattitude and longtitude of point i in the path.

Write a program `gps_trail_vector.cc` that let's the user add points to a GPS path object. After the user is finished entering coordinates, the program should print the path and also calculate and print its total distance.

Hints:

- The program is similar to the phone book example from the lecture code in the `src/phonebook` directory.
- Google Maps measures the total distance of paths if you click on multiple points while in the "Measure distance" mode.
- Your code needs to be aligned properly and you must use consistent capitalization in variable and function names. Homeworks with inconsistent naming and indentation will lose up-to 15 points.

Assignment 2 GPS Paths with Linked Lists

Write a structure `GPSCoordNode` that stores the longtitude and lattitude of a GPS coordinate and a `next` pointer of type `GPSCoordNode *`.

Write a class `GPSPathList` with the same functionality as `GPSPath` that stores the `GPSCoordNode` objects as a linked list.

Write a program in `gps_trail_list.cc` that uses `GPSPathList` to perform the same operations as `gps_trail_vector.cc`.

Hints:

- The program is similar to the phone book example from the lecture code in the `src/phonebook_with_lists` directory.
- If you do things right, `gps_trail_list.cc` and `gps_trail_vector.cc` will be almost the same.
- Your code needs to be aligned properly and you must use consistent capitalization in variable and function names. Homeworks with inconsistent naming and indentation will lose up-to 15 points.

NOTE: You have two extra days at the weekend for this homework. Understanding the phone book examples in the class and working on this homework should really help you with C/C++ and data structure basics.