Hack 2.0

Computer Science I

Department of Computer Science & Engineering
University of Nebraska–Lincoln

Introduction

Hack session activities are small weekly programming assignments intended to get you started on full programming assignments. You may complete the hack on your own, but you are *highly encouraged* to work with another student and form a hack pair. Groups larger than 2 are not allowed. However, you may discuss the problems *at a high level* with other students or groups. You may not share code directly.

If you choose to form a Hack Pair, you *must*:

1. Both join a hack pair on Canvas (go to People then Hack Pairs)
2. You must both work on the hack equally; it must be an equal effort by both partners. Do not undermine your partner's learning opportunity and do not undermine your own by allowing one parter to do all the work.
3. Turn in only one copy of the code under the individual whose last name comes first (with respect to Canvas).

You are graded based on style, documentation, design and correctness. For detail, see the general course rubric.

Category	Point Value
Style	2
Documentation	2
Design	5
Correctness	16
Total	**25**

Table 1: Rubric

Correctness: points will be awarded proportionally per test case.

Problem Statement

Consider two locations, an origin and a destination, on the globe identified by their latitude and longitude. The distance between these two locations can be computed using the Spherical Law of Cosines. In particular, the distance d is

$$d = \arccos\left(\sin(\varphi_1)\sin(\varphi_2) + \cos(\varphi_1)\cos(\varphi_2)\cos(\Delta)\right) \cdot R$$

where

- φ_1 is the latitude of location A, φ_2 is the latitude of location B
- Δ is the difference between location B's longitude and location A's longitude
- R is the (average) radius of the earth, 6,371 kilometers

Write a program that *prompts* the user to enter the latitude and longitude of two locations and then computes the distance between them using the above formula. Note that latitude inputs will be in degrees and in the range $[-90, 90]$ and longitude will be in degrees in the range $[-180, 180]$. Negative values correspond to the western and southern hemispheres.

Note that the formula above assumes that latitude and longitude are measured in radians r, $-\pi \leq r \leq \pi$. You can convert from degrees deg to radians r using the formula

$$r = \frac{deg}{180} \cdot \pi$$

Your output should look something like the following.

```
Location Distance
========================
Origin:      (41.948300, -87.655600)
Destination: (40.820600, -96.705600)
Air distance is 764.990931 kms
```

Instructions

- You are encouraged to collaborate any number of students before, during, and after your scheduled hack session.
- Design at least 3 test cases *before* you begin designing or implementing your program. Test cases are input-output pairs that are known to be correct using means other than your program.

- Include the name(s) of everyone who worked together on this activity in your source file's header.
- Name your program `airDistance.c`, and turn it in via webhandin, making sure that it runs and executes correctly in the webgrader. Each individual student will need to hand in their own copy and will receive their own individual grade.
- Remember to RTM (Read The Manual) on the math library to see which function(s) you may find useful and how to use them.
- Depending on your compiler/system configuration you *may* need to use the `-lm` flag to link in the math library when compiling. For example:

  ```
  gcc -lm airDistance.c
  ```

  or in some systems (ubuntu/CS50):

  ```
  gcc airDistance.c -lm
  ```