EXAM 3

ENGINEERING 200

FALL, 2017

(100 points)

(Due: November 11, 2017 at 11:59 p.m. CST)

**NOTE: Your program must use the equations provided in this exam. Do not use equations that you may find on the internet or any other source. Do not substitute equations. Also, do not place one equation inside another equation.**

**INTRODUCTION:**

The Global Position System (GPS) was originally developed for the U.S. military. It uses 24 satellites that circle the Earth at an altitude of 12,427 miles, and the satellites complete two full orbits every day. The satellites use an atomic clock that is accurate to within one second every 70,000 years. The GPS system determines a location on the Earth in latitude and longitude coordinates. When latitude and longitude coordinates are known for two locations, a great circle distance can be computed between the two locations.

To compute a great circle distance between two locations that are specified by latitude and longitude in the Northern Hemisphere, a series of coordinate transformations are performed as follows:

* In the main function, compute the rectangular coordinates (x,y,z) from the latitude and longitude coordinates for the two locations. Note: ρ is rho, ϕ is phi, and θ is theta.

First location:

ρ = Radius of the Earth in miles.

ϕ1 = (90°– first latitude)

θ1 = (360°– first longitude)

Second location:

ϕ2 = (90°– second latitude)

θ2 = (360°– second longitude)

* In the user-defined function, compute the angle (gamma) between the two vectors (x1,y1,z1) and (x2,y2,z2) using the following four equations:

Note: The program must use the **power** and **square root** built-in functions. These four equations must be inside the user-defined function.

* In the main function, compute the great circle distance as follows:

In the preprocessor directive section, define pi and Earth radius as symbolic variables. To obtain the most accurate computations, for pi use 3.141592654 and for the Earth radius use 3958.89 (this is miles).

The input file called **cities** contains: 1) the number of cities; 2) the names of the cities; 3) the latitude (degrees and minutes); 4) the hemisphere; 5) the longitude (degrees and minutes); 6) the direction.

**EXAM:**

Using an **array of structures**, write a C program that will compute the great circle distance between a pair of cities. For example, when looking at the initial order of the cities in the input file the pairing would be Montreal and London, Chicago and Los Angeles, Winnipeg and Honolulu. An incorrect pairing would be Montreal and London, London and Chicago, Chicago and Los Angeles, Los Angeles and Winnipeg, Winnipeg and Honolulu. Note that once the sorting has been performed, the city pairing will be different than the above example.

In your defined structure, the structure members must be: 1) a character array for the city name; 2) variables for the latitude and longitude values (degrees and minutes); 3) a character variable for the hemisphere; 4) a character variable for the direction.

Your program will:

* Use a “for” loop to read all the data from the input file into the array of structures.
* Use a second “for” loop to sort the data by latitude degrees such that the smallest latitude is at the bottom of the array and the largest latitude is at the top of array.
* Use a third “for” loop to control printing the pair of cities, the computations, calling your user-defined function that computes the angle gamma, computing the great circle distance, and printing the computed distance.

After computing the x, y, and z coordinates for a pair of cities, the main function will call a user-defined function that will compute the dot product, distance 1, distance 2, and the angle gamma. The user-defined function will return gamma to the main function, and the main function will compute the great circle distance.

Your program will print to the computer screen and to an output file called **great\_circle\_distance**.

**OUTPUT FORMAT:**

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

GREAT CIRCLE DISTANCE CALCULATIONS

LATITUDE LONGITUDE

CITY Deg Min Deg Min

ssssssssssss xxx xx c xxx xx c

Note: The **ssssssssssss** means you are printing a string of characters. The **c** means you are printing a character.

ssssssssssss xxx xx c xxx xx c

Distance between the two cities = xxxx.x miles

ssssssssssss xxx xx c xxx xx c

ssssssssssss xxx xx c xxx xx c

Distance between the two cities = xxxx.x miles

ssssssssssss xxx xx c xxx xx c

ssssssssssss xxx xx c xxx xx c

Distance between the two cities = xxxx.x miles

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

**FILE PATHS:**

Before submitting your program to Blackboard, you must set the following file paths in your program.

"u:\\engr 200\\cities.txt"

"u:\\engr 200\\great\_circle\_distance.txt"

**OTHER INFORMATION:**

1. Your program must follow the same format used for all programming assignments from the beginning of the

semester, and Exam #1. That is to say, it must have a title block, a program description, and a list of

variables. See the example shown below. Also, within the program there must be proper indenting and

comments relating to major sections of the program like those presented in the demo programs.

/\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

ENGR 200-XX DATE: mm/dd/yyyy

EXAM: #3 Author:

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

PROGRAM DESCRIPTION:

This program ...

DESCRIPTION OF VARIABLES

NAME | TYPE | DESCRIPTION

--------------------------------------------------------------------------------

pi | symbolic | mathematical constant

. . .

. . .

. . .

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*/

/\* Preprocessor directives \*/

2. Before submitting to Blackboard, name your program **exam\_3\_firstname\_lastname**.

3. Submit your exam program to Blackboard using standard procedures.

4. Grading will be based on, but not limited to, the following criteria:

* Program does not have a title block, program description, and list of variables, or they are incomplete, or inappropriate. Program does not use programming indentation for readability.
* Program does not print output according to the output format given on page 2.
* Program does not incorporate the required file paths as shown on page 2.
* Program does not use a defined array of structures.
* Program does not use equations given in the exam.
* Program does not use a “for” loop to read all data.
* Program does not us a second “for” loop to sort the data.
* Program does not use a third “for” loop to compute and print the output.
* Program does not have a user-defined function to compute the dot product, distance 1, distance 2, and gamma, and return the computed gamma to the main function.
* Program does not read all data into the array before computations are performed.
* Program compiles and runs, but the computations are incorrect.
* Program compiles, but will not run.
* Program does not compile.