ENGINEERING H192
DAILY ASSIGNMENT B15

Throughout the quarter you have used functions such as **printf()** and **pow()** which are located in already existing libraries like **stdio.h** and **math.h**. In this assignment, you will create your own library consisting of defined constants and some functions that you have written or will write today.

A. Create a header file called **mylib.h** that contains definitions for the constants:

PI the value of pi to at least 7 places
RAD_E the (average) radius of the Earth: 6,372.80 km

and the following function prototypes:

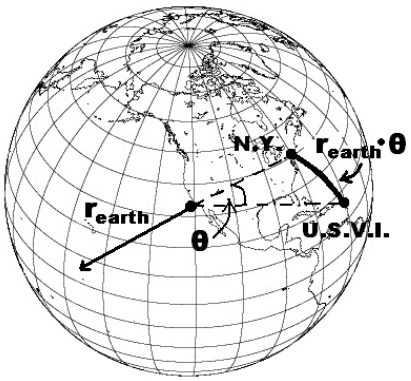
```
float mydot (float a[], float b[]);
void mycross (float a[], float b[], float *c);
void myvect_mag (float a[], float *b);
```

B. Create and compile **(only)** a library file called **mylib.cpp** that contains the two user-written functions, **mydot( )** and **mycross( )**, from **b14.cpp**. You must write another function, **myvect_mag( )** that computes the **magnitude** of a vector:

$$|\boldsymbol{A}| = \left[A_x^2 + A_y^2 + A_z^2\right]^{1/2}$$

As shown, a vector's magnitude is represented by the vector name in absolute value brackets. Your compiled file will be named **mylib.o**. The "compile only" command is: > **g++ mylib.cpp -c**

C. Test your library by writing **b15.cpp** containing the pre-processor directive **#include "mylib.h".** This program will (approximately) calculate the distance between any two locations on Earth, based on their latitude and longitude coordinates. As shown in the figure to the right, an **arc length** will be calculated based on the angle, **Θ**, between vectors from the center of the Earth to the locations of interest using the following relation:



$$cos(\theta) = \frac{\boldsymbol{A} \cdot \boldsymbol{B}}{|\boldsymbol{A}||\boldsymbol{B}|}$$

Each latitude and longitude coordinate vector will be of the form [**latitude**, **longitude**, **0**]. The z-component of the vector, altitude, is assumed to be zero for this calculation. Coordinates must be converted from degrees and minutes into decimal form, i.e. 47° 31' E becomes 47 + 31/60, or 47.52. Also, a latitude in the Southern Hemisphere or a longitude in the Western Hemisphere will be negative in value for this calculation.

Multiply **Θ** (the **acos( )** function in <**math.h**> should be useful) by Earth's radius to determine an estimate for the distance between the two points. Determine, for comparison, what fraction this distance is of Earth's circumference and convert this value to a percentage.

When complete, **b15.cpp** should prompt the user for the decimal coordinates of two points, use the **mydot( )** and **myvect_mag( )** functions and defined constants from **"mylib.h"** to make necessary calculations, and print the distance between the points and the calculated percentage to the screen. The compile/link command will be: > **g++ -o b15.out b15.cpp mylib.o**

When your library and test program are working correctly, print **mylib.h**, **mylib.cpp** and **b15.cpp** and submit them with this sheet.

Name_____Instructor_____Seat_____Hour_____