

CS205 C/ C++ Programming - Lab Assignment Template

Name: 张佳晨(Jiacheng ZHANG)

SID: 11713020

Part 1 - Analysis

Get the names, the latitudes and the longitudes of the two cities, which will be given from `stdin`.

Assume the Earth is a perfect sphere. Let all angles be measured in signed degrees (negative latitude means South, negative longitude means West)

$$\begin{aligned} \phi &= 90 - \text{latitude} \\ \theta &= \text{longitude} \end{aligned}$$

Let the angles for the two points be (ϕ_1, θ_1) and (ϕ_2, θ_2) . Then compute

$$c = \sin(\phi_1) * \sin(\phi_2) * \cos(\theta_1 - \theta_2) + \cos(\phi_1) * \cos(\phi_2)$$

Then the shortest great circle distance between the two points is

$$d = R * \text{Arccos}(c) * \pi / 180$$

the `sin()`, `cos()`, `Arccos()` will be given in `<math.h>`

Part 2 - Code

```
#include <stdio.h>
#include <math.h>

double compute(double, double, double, double);
double latitude_1 = -200;
double longitude_1 = -200;
double latitude_2 = -200;
double longitude_2 = -200;
char city_1[100];
char city_2[100];
void read(void);

int main(){
/* TODO (#1#): read the loction */
    read();
/*
    Let phi = 90 - latitude.
    Let theta = longitude
*/
    double phi_1 = 90 - latitude_1;
    double theta_1 = longitude_1;
    double phi_2 = 90 - latitude_2;
    double theta_2 = longitude_2;
/* print the result */
    double distance = 0.0;
    distance = compute(phi_1, theta_1, phi_2, theta_2);
    printf("The distance between %s and %s is %f km", city_1, city_2, distance);

    return 0;
}

double compute(double phi_1, double theta_1, double phi_2, double theta_2){
    double Pi = acos(-1.0);
    double c = 0.0;
    phi_1 = phi_1/180*Pi;
    theta_1 = theta_1/180*Pi;
    phi_2 = phi_2/180*Pi;
    theta_2 = theta_2/180*Pi;
    c = sin(phi_1)*sin(phi_2)*cos(theta_1-theta_2) + cos(phi_1)*cos(phi_2);
/* The radius of the earth is R = 6371 km*/
    int R = 6371;
    double d = 0.0;
    d = R*acos(c);
    return d;
}

void read(){
    do{
        printf("Please input the name of the first city, then press ENTER\n");
        fflush(stdin);
        gets(city_1);
        printf("Please input the name, latitude and longitude of the first city, divided with space:\n");
        fflush(stdin);
        scanf("%lf",&latitude_1);
        scanf("%lf",&longitude_1);
        if(!(-90 <= latitude_1 && latitude_1 <= 90)||!(-180 <= longitude_1 && longitude_1 <= 180)
||latitude_1 == -200||longitude_1 == -200){
            printf("\n\n\n*****The data is invalid, please input again*****\n\n\n");
        }
    }while(!(-90 <= latitude_1 && latitude_1 <= 90)||!(-180 <= longitude_1 && longitude_1 <= 180)
||latitude_1 == -200||longitude_1 == -200);
    do{
        printf("Please input the name of the second city, then press ENTER\n");
        fflush(stdin);
        gets(city_2);
        printf("Please input the name, latitude and longitude of the second city, divided with space:\n");
        fflush(stdin);
        scanf("%lf",&latitude_2);
        scanf("%lf",&longitude_2);
        if(!(-90 <= latitude_2 && latitude_2 <= 90)||!(-180 <= longitude_2 && longitude_2 <= 180)
||latitude_2 == -200||longitude_2 == -200){
            printf("\n\n\n*****The data is invalid, please input again*****\n\n\n");
        }
    }while(!(-90 <= latitude_2 && latitude_2 <= 90)||!(-180 <= longitude_2 && longitude_2 <= 180)
||latitude_2 == -200||longitude_2 == -200);
}
```

Part 3 - Result & Verification

Test case #1:

```
Input:
Beijing
39.9139 116.3917
Shenzhen
22.55 114.1
Output: The distance between Beijing and Shenzhen is 1942.835731 km
```

Test case #2:

```
Input:
New York, USA
40.7127 -74.0059
San Francisco, USA
37.7833 -122.4167
Output: The distance between New York, USA and San Francisco, USA is 4128.553030 km
```

Part 4 - Difficulties & Solutions

The prompt we made may cause the error when reading, so `fflush(stdin)` is used.

The location that the user input may be invalid , so we use this block of codes to check and solve this problem.

```
do{
    /* read the data */

    /* check the data */
    if(!(-90 <= latitude_2 && latitude_2 <= 90)||!(-180 <= longitude_2 && longitude_2 <= 180)
        ||latitude_2 == -200||longitude_2 == -200){
        printf("\n\n*****The data is invalid, please input again*****\n\n\n");
    }
}while(!(-90 <= latitude_2 && latitude_2 <= 90)||!(-180 <= longitude_2 && longitude_2 <= 180)
        ||latitude_2 == -200||longitude_2 == -200);
```