

CS205 C/ C++ Programming Assignment 1

Name: 傅伟堡(Weibao Fu)

SID:11812202

Part 1-Analysis

The problem is to calculate the flying distance of two city according to their latitude and longitude, if we assume the Earth is a perfect sphere, then we can calculate the distance between them directly by the equation below. (Earth radius: 6371km)

$$\begin{aligned} \text{let } \phi &= 90 - \text{latitude}, \theta = \text{longitude} \\ c &= \sin(\phi_1) * \sin(\phi_2) * \cos(\theta_1 - \theta_2) + \cos(\phi_1) * \cos(\phi_2) \\ d &= R * \arccos(c) \end{aligned}$$

Part 2-Code

```
#include<stdio.h>
#include<Math.h>
#include<cstring>
using namespace std;

const int R = 6371;
#define PiDiv (0.017453292519943296) //π/180
#define DegToRad(x) ((x)*PiDiv) // transfer Degree to Radian

int main(){
    char city1[100] , city2[100] , temp[1000];
    double lati1 , lati2 , longi1 , longi2;
    bool state;
    int tem;
    do{
        state = false;
        printf("The first city:");
        scanf("\n");
        gets(city1);
        tem=strlen(city1)-1;
        while(city1[tem--]!=' ');
```

```

        city1[tem+2]='\0';
        for(int i=0;i<strlen(city1);i++){
            if(strlen(city1)>=30||!((city1[i]!=64&&city1[i]>=32&&city1[i]<=90)||
(city1[i]>=97&&city1[i]<=122)||city1[i]==' '||city1[i]==' ')){
                printf("The name's format is incorrect.\n");
                state=true;
                break;
            }
        }
    }while(state);
    do{
        state = false;
        printf("The latitude and longitude of first city:");
        if(!scanf("%lf%lf", &lati1 ,
&longi1)||lati1<-90||lati1>90||longi1<-180||longi1>180){
            printf("The value of latitude and longitude are incorrect.\n");
            state = true;
            gets(temp);
            continue;
        }
        gets(temp);
        for(int i=0;i<strlen(temp);i++){
            if(temp[i]!=' '){
                printf("The value of latitude and longitude are incorrect.\n");
                state = true;
                break;
            }
        }
    }while(state);
    do{
        state = false;
        printf("The second city:");
        scanf("\n");
        gets(city2);
        tem=strlen(city2)-1;
        while(city2[tem--]!=' ');
        city2[tem+2]='\0';
        for(int i=0;i<strlen(city2);i++){
            if(strlen(city2)>=30||!((city2[i]!=64&&city2[i]>=32&&city2[i]<=90)||
(city2[i]>=97&&city2[i]<=122)||city2[i]==' '||city2[i]==' ')){
                printf("The name's format is incorrect.\n");
                state=true;
                break;
            }
        }
    }while(state);
    do{
        state = false;
        printf("The latitude and longitude of second city:");
        if(!scanf("%lf%lf", &lati2 ,
&longi2)||lati2<-90||lati2>90||longi2<-180||longi2>180){
            printf("The value of latitude and longitude are incorrect.\n");
            state = true;
            gets(temp);
            continue;
        }
        gets(temp);
        for(int i=0;i<strlen(temp);i++){

```

```

        if(temp[i]!=' '){
            printf("The value of latitude and longitude are incorrect.\n");
            state = true;
            break;
        }
    }
}while(state);

double phi1 = DegToRad(90 - lati1);
double phi2 = DegToRad(90 - lati2);
double theta = DegToRad(longi1-longi2);
double c = sin(phi1)*sin(phi2)*cos(theta)+cos(phi1)*cos(phi2);
double d = R*acos(c);
printf("The distance between %s and %s is %.1f km.\n",city1,city2,d);
return 0;
}

```

Part 3-Result & Verification

Test case #1

Input: Shenzhen
 22.55 114.1
 Beijing
 39.9139 116.3917
 Output: The distance between Shenzhen and Beijing is 1942.8 km.

```

The first city:Shenzhen
The latitude and longitude of first city:22.55 114.1
The second city:Beijing
The latitude and longitude of second city:39.9139 116.3917
The distance between Shenzhen and Beijing is 1942.8 km.

-----
Process exited after 0.5348 seconds with return value 0
请按任意键继续. . .

```

Test case #2

Input: Kolkata, India
 22.567 88.367
 Sydney, Australia
 -33.865 151.209444
 Output: The distance between Kolkata, India and Sydney, Australia is 9137.5 km.

```

The first city:Kolkata, India
The latitude and longitude of first city:22.567 88.367
The second city:Sydney, Australia
The latitude and longitude of second city:-33.865 151.209444
The distance between Kolkata, India and Sydney, Australia is 9137.5 km.

-----

Process exited after 0.5931 seconds with return value 0
请按任意键继续. . . █

```

Test case #3

```

Input:  New York, USA
        40.7127 -74.0059
        London, UK
        51.5072 -0.1275
Output: The distance between New York, USA and London, UK is 5570.2 km.

```

```

The first city:New York, USA
The latitude and longitude of first city:40.7127 -74.0059
The second city:London, UK
The latitude and longitude of second city:51.5072 -0.1275
The distance between New York, USA and London, UK is 5570.2 km.

-----

Process exited after 8.74 seconds with return value 0
请按任意键继续. . . █

```

Test case #4

```

The first city:Tooouooooooooooooooooooooooooooooooooooooooooooooo long
The name's format is incorrect.
The first city:input error 00,[[[]]]
The name's format is incorrect.
The first city:                                London, UK
The latitude and longitude of first city:999 -888
The value of latitude and longitude are incorrect.
The latitude and longitude of first city:51.5072 -0.1275
The second city:                                Paris, France
The latitude and longitude of second city:48.8567 2.3508
The distance between London, UK and Paris, France is 343.5 km.

-----

Process exited after 100.9 seconds with return value 0
请按任意键继续. . . █

```

Part 4-Difficulties & Solutions

- 1.Using scanf() cause we cannot read the city's name completely, so gets() is used.
- 2.Since the function sin() and cos() require a radian rather than a degree, we define a macro to transfer the degree to the radian.

3. Since the distance is a floating-point number, we set its type is double when we define it. And we choose to reserve a decimal when output it.

4. We should print prompt information when user enter the wrong format, so we give a judgement when we input the city's name, latitude and longitude. If the information is incorrect, the program will remind user input the information again.

5. If the user input spaces before or behind the city's name, we should output the city's name without spaces, so we choose to `scanf("\n")` before we read the city's name, and judge the amount of the spaces in the end, delete these spaces.

6. Since we choose to use `scanf()` to read the latitude and longitude, if we don't take measure, the program will crash when the user input incorrect format when the program prompt the user input latitude and longitude. The solution is that we use `gets()` to read the remaining data, and judge if the data contains others information, if yes, prompt the user input the right format information again.