CS205 C/ C++ Programming - Lab Assignment 1

Name: 邱煜（Qiu Yu）

SID: 11611127

Part1 - Analysis

The problem is to calculate the flying distance between two cities.

I use `prinf()` to print the prompt messages to guide the users to input the two cities' names, longtitudes and latitudes. And use `scanf()` to receive the input and use regular expression to deal with the situations that some cities' names contains space. And then I use `getchar()` to flush the `return`. The input cities' name should **<= 30**, if > 30, this program will show a message and need the user to input again. Also, the latitude should be in range **[-90, 90]** and the longtitude should be in range **[-180, 180]**, if the input is not suitable, the user will be guided to input again until satifsy the requirements.

The function I use to compute the distance is as below

> phi = 90 - latitude
>
> theta = longitude
>
> c = sin(phi1 * pi / 180) * sin(phi2 * pi / 180) * cos((theta1-theta2)* pi / 180) +cos(phi1 * pi / 180) * cos(phi2 * pi / 180)
>
> d = R*arccos(c)

Part2 - Code

```
1  //
2  //  main.c
3  //  lab1
4  //
5  //  Created by 邱煜 on 2019/2/22.
6  //  Copyright © 2019 邱煜. All rights reserved.
7  //
8
9  #include <stdio.h>
10 #include <math.h>
11 #include <string.h>
12 #include <stdlib.h>
```

```c

#define Pi 3.141592654
#define R 6371

double calculate(double fLati, double fLong, double sLati, double sLong);
double convert(double angle);

int main() {
    char fCity[100], sCity[100];
    double fLati, fLong, sLati, sLong;
    int response;
    double result;

    printf("Please input the first city's name: ");
    scanf("%[^\n]", fCity);
    getchar();
    while (strlen(fCity)>30) {
        printf("Input is too long, please do not exceed 30 characters:
");
        scanf("%[^\n]", fCity);
        getchar();
    }

    printf("Please input the first city's latitude: ");
    response = scanf("%lf", &fLati);
    getchar();
    while (!(fLati >= -90 && fLati <= 90 && response == 1)) {
        printf("The latitude should be in range [-90, 90]: ");
        scanf("%lf", &fLati);
        getchar();
    }

    printf("Please input the first city's longitude: ");
    response = scanf("%lf", &fLong);
    getchar();
    while (!(fLong >= -180 && fLong <= 180 && response == 1)) {
        printf("The longtitude should be in range [-180, 180]: ");
        scanf("%lf", &fLong);
        getchar();

    }

    printf("Please input the second city's name: ");
    scanf("%[^\n]", sCity);
    getchar();
    while (strlen(sCity)>30) {
        printf("Input is too long, please do not exceed 30 characters:
");
        scanf("%[^\n]", sCity);
        getchar();
    }

    printf("Please input the second city's latitude: ");
    response = scanf("%lf", &sLati);
    getchar();
    while (!(sLati >= -90 && sLati <= 90 && response == 1)) {
        printf("The latitude should be in range [-90, 90]: ");
        scanf("%lf", &sLati);
        getchar();
```

```
    }

    printf("Please input the second city's longitude: ");
    response = scanf("%lf", &sLong);
    getchar();
    while (!(sLong >= -180 && sLong <= 180 && response == 1)) {
        printf("The longitude should be in range [-180, 180]: ");
        scanf("%lf", &sLong);
        getchar();
    }

    result = calculate(fLati, fLong, sLati, sLong);
    printf("The distance between %s and %s is %f km.\n", fCity, sCity,
result);
    return 0;
}


double calculate(double fLati, double fLong, double sLati, double sLong){
    // 0 <= phi <= 180 -180 <= theta <= 180
    // -90 <= latitude <= 90 -180 <= longtitude <= 180
    double phi1, phi2, theta1, theta2;
    double c;
    double result;
    phi1 = 90 - fLati;
    theta1 = fLong;
    phi2 = 90 - sLati;
    theta2 = sLong;
    c = sin(convert(phi1)) * sin(convert(phi2)) * cos(convert(theta1-
theta2)) + cos(convert(phi1)) * cos(convert(phi2)) ;
    result = R * acos(c);
    return result;
}

double convert(double angle){
    // conversion from angle to radian
    return angle * Pi / 180;
}

```

#2 Part 3 - Result & Verification

Test case #1:

```
Input:
    Shenzhen 22.55 114.1
    Beijing 39.9139 116.3917
Output:
    1942.835731
Verification:
    1950
```

```
Personal@Coreys ~/documents/c_workspace/course/lab1/lab1 master ●+ ./a.
out
Please input the first city's name: Shenzhen
Please input the first city's latitude: 22.55
Please input the first city's longitude: 114.1
Please input the second city's name: Beijing
Please input the second city's latitude: 39.9139
Please input the second city's longitude: 116.3917
The distance between Shenzhen and Beijing is 1942.835731 km.
```

Travel Route:		
Shenzhen Arpt, Shenzhen, CN (SZX) to		
Beijing Airport, Beijing, CN (BJS)		
One Way Distances:		
Distance	1210 miles	1950 km
Elite bonus	0 miles	0 km
Class of service bonus	0 miles	0 km
Special promotion bonus	0 miles	0 km
TOTAL	1210 miles	1950 km

Test case #2:

```
Input:
    New York, USA 40.7127 -74.0059
    Rio de Janeiro, Brazil -22.9083 -43.1964
Output:
    7758.185989
Verification:
    7760
```

```
Personal@Coreys ~/documents/c_workspace/course/lab1/lab1 master ●+ ./a.
out
Please input the first city's name: New York, USA
Please input the first city's latitude: 40.7127
Please input the first city's longitude: -74.0059
Please input the second city's name: Rio de Janeiro, Brazil
Please input the second city's latitude: -22.9083
Please input the second city's longitude: -43.1964
The distance between New York, USA and Rio de Janeiro, Brazil is 7758.185989 km.
```

Travel Route:		
New York City Airport, New York City, NY (NYC) to		
Rio De Janeiro Airport, Rio De Janeiro, BR (RIO)		
One Way Distances:		
Distance	4820 miles	7760 km
Elite bonus	0 miles	0 km
Class of service bonus	0 miles	0 km
Special promotion bonus	0 miles	0 km
TOTAL	4820 miles	7760 km

#2 Part 4 - Difficulties & Solutions

---

1. Some cities' names contain spaces such as **New York**, I use regular expression to solve this problem. For example: `scanf("%[^\n]", fCity);`
2. And because this program has many input, so I need to use `getchar()` to flush the input cache after each input finished.
3. Need to deal with the situations that the input latitude or longtitude is invalid. I use the return value of `scanf()` to determine if the value is valid.