CS205 C/ C++ Programming - Lab Assignment Template

Name: 周民涛(zhou mintao)
SID: 11912725
environment:
gcc (GCC) 7.4.0 visual Studio Code

part1 - Analysis

> 1. 根据题意我们得知，我们需要计算出两座城市之间的距离
> 2. 题目输入两座城市名和经纬度信息，输出距离，可以用sturct类来存储城市的相关信息
> 3. 当出现异常时，我们需要爆出异常的具体信息，并退出程序
> 4. 城市名不能包括特殊字符,所以我们需要一个函数进行判断judge_city_name(string name)
> 5. 由于输入风格不确定，这里需要我们分类进行讨论
> 6. 经纬度输入的格式以及范围需要正确，所以需要judge_index_format(string a,string b)和judge_city_index(double latitude,double longitude)函数辅助判断
> 7. 计算距离所需要的公式，在作业文案中已给出，需要我们用代码进行实现findDistance(city city1, city city2)
> 8. 同时我们考虑一些特殊的情况如(0,0)这个地方是没有城市的

part 2 - Code

```cpp
#include<iostream>
#include<cmath>
#include<string>
#include<regex>

using namespace std;

const double R = 6371;
const double PI = 3.1415926535;
const double val = PI/180;

struct city{
    string name;
    double latitude;
    double longitude;
};


double findDistance(city city1, city city2);
string remove_space(string name);
int judge_city_index(double latitude,double longitude);
int judge_city_name(string name);
int judge_index_format(string a,string b);
bool check_double_char(string & str);
int some_special_case(double a, double b);


int main(){
    //difine the city
    city c1 ;
    city c2 ;
    //input the data
    cout<<"please input the first city name(without special characters):";
    char char1;
    while((char1=cin.get())!='\n')
    c1.name=c1.name+char1;
    c1.name =remove_space(c1.name);
    if(judge_city_name(c1.name)){
        return -1;
    }
    cout<<"please input the latitude and longitude:";
    string lat1;
    string lon1;
    cin >> lat1;
    cin >> lon1;
    if(judge_index_format(lat1,lon1)){
        return -1;
    }
    c1.latitude = stod(lat1);
    c1.longitude = stod(lon1);
```

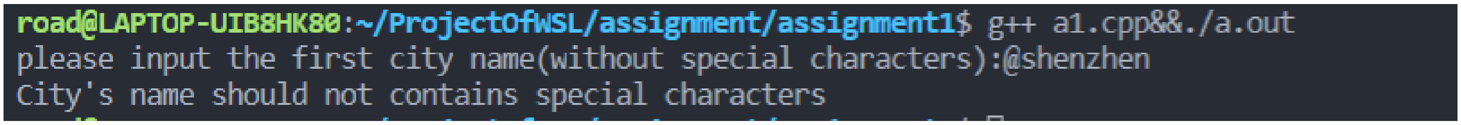
```
 if(judge_city_index(c1.latitude,c1.longitude)){
    return -1;
 }

 if(some_special_case(c1.latitude,c1.longitude)){
     return -1;
 }
 cin.get();

 cout<<"please input the second city name(without special characters):";
 char char2;
 while((char2=cin.get())!='\n')
 c2.name=c2.name+char2;
 c2.name =remove_space(c2.name);
 if(judge_city_name(c2.name)){
     return -1;
 }

cout<<"please input the latitude and longitude:";
 string lat2;
 string lon2;
 cin >> lat2;
 cin >> lon2;

 if(judge_index_format(lat2,lon2)){
     return -1;
 }

 c2.latitude = stod(lat2);
 c2.longitude = stod(lon2);
 cin.get();

 if(judge_city_index(c2.latitude,c2.longitude)){
    return -1;
 }

 if(some_special_case(c2.latitude,c2.longitude)){
     return -1;
 }



 //calu the distance
 double distance = findDistance(c1,c2);
 //out
 cout<<endl;
 cout << "The first city:"<<c1.name<<endl;
 cout << "The latitude and longitude of first city:";
 cout<< c1.latitude;
 cout<<" " ;
 cout<< c1.longitude<<endl;
```

```
    cout << "The second city:"<<c2.name<<endl;
    cout << "The latitude and longitude of second city:";
    cout<< c2.latitude;
    cout<<" ";
    cout<< c2.longitude<<endl;

    cout<< "The distance is: "<< distance <<" km"<<endl;
    return 0;
}

double findDistance(city city1, city city2){
    double ans =0;
    double phi1 = 90 - city1.latitude;
    double theta1 = city1.longitude;
    double phi2 = 90 - city2.latitude;
    double theta2 = city2.longitude;
    double c = sin(phi1*val)*sin(phi2*val)*cos((theta1-theta2)*val)+cos(phi1*val)*cos(phi2*val);
    ans = R * acos(c);
    return ans;
}
string remove_space(string name){
    if(name.empty()){
        return name;
    }else{
        name.erase(0,name.find_first_not_of(" "));
        name.erase(name.find_last_not_of(" ")+1);
        return name;
    }
}
int judge_city_index(double latitude,double longitude){
    if(!(latitude<=90&&latitude>=-90)||!(longitude<=180&&longitude>=-180)){
        cout<<"latitude should be [-90,+90]"<<endl;
        cout<< "longitude should be [-180,+180]"<<endl;
        return -1;
    }
    return 0;
}
int judge_city_name(string name){
    for(auto i:name){
        if(!(' '==i ||','==i ||(i>='A' &&i<= 'Z')||(i>='a' &&i<='z'))){
            cout << "City's name should not contains special characters" <<endl;
            return -1;
        }
    }
    return 0;
}
int judge_index_format(string a,string b){
    for(auto i:a){
        if(!('.'==i ||(i>='0' &&i<= '9') || i == '+' || i=='-')){
            cout<<"Invalid format, not contains special chararcters."<<endl;
           return -1;
        }
    }
     for(auto i:b){
        if(!('.'==i ||(i>='0' &&i<= '9') || i == '+' || i=='-')){
             cout<<"Invalid format, not contains special chararcters."<<endl;
            return -1;
        }
    }
    if(!check_double_char(a)||!check_double_char(b)){
        return -1;
    }

    try {
        stof(a);
        stof(b);
    } catch(std::invalid_argument e) {
        cout << "input should follow correct double format" << endl;
        return -1;
    }

    return 0;
}
bool check_double_char(string & str) {
    if (count(str.begin(), str.end(), '.') > 1 || (count(str.begin(), str.end(), '+') + count(str.begin(), str.en
        cout<< "Invallid, too many '.' '+' '-'."<<endl;
        return false;
    }
    return true;
}
int some_special_case(double a, double b){
    if(a==0.0&&b==0.0){
        cout<<"This index not exist a city."<<endl;
        return -1;
    }
    return 0;
}
```

part 3 - Result & Verification

Test case #1: **非法城市名**

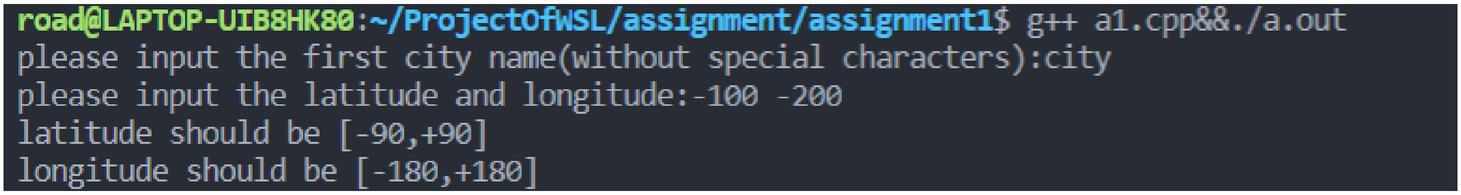
```
input:
@shenzhen
output:
City's name should not contains special characters
```



```
road@LAPTOP-UIB8HK80:~/ProjectOfWSL/assignment/assignment1$ g++ a1.cpp&&./a.out
please input the first city name(without special characters):@shenzhen
City's name should not contains special characters
```

Test case #2: **经纬度越界**

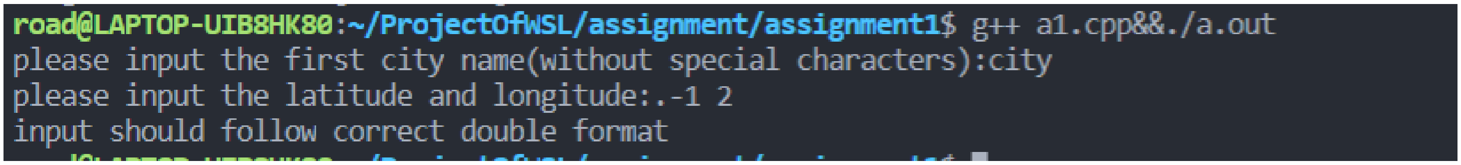
```
input:
city
-100 -200
output:
latitude should be [-90,+90]
longitude should be [-180,+180]
```



```
road@LAPTOP-UIB8HK80:~/ProjectOfWSL/assignment/assignment1$ g++ a1.cpp&&./a.out
please input the first city name(without special characters):city
please input the latitude and longitude:-100 -200
latitude should be [-90,+90]
longitude should be [-180,+180]
```

Test case #3: **经纬度格式不对**

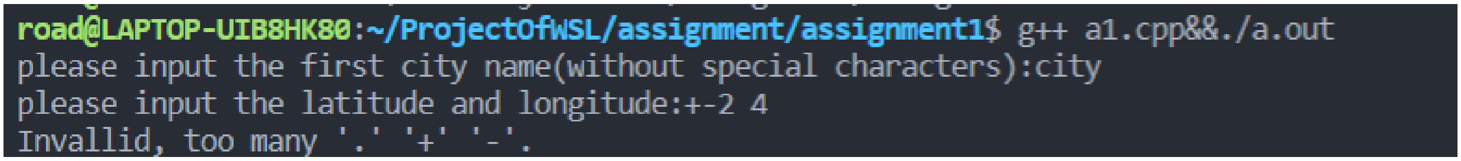
```
input:
city
.-1 2

output:
input should follow correct double format
```



```
road@LAPTOP-UIB8HK80:~/ProjectOfWSL/assignment/assignment1$ g++ a1.cpp&&./a.out
please input the first city name(without special characters):city
please input the latitude and longitude:.-1 2
input should follow correct double format
```

Test case #4: **经纬度格式不对**

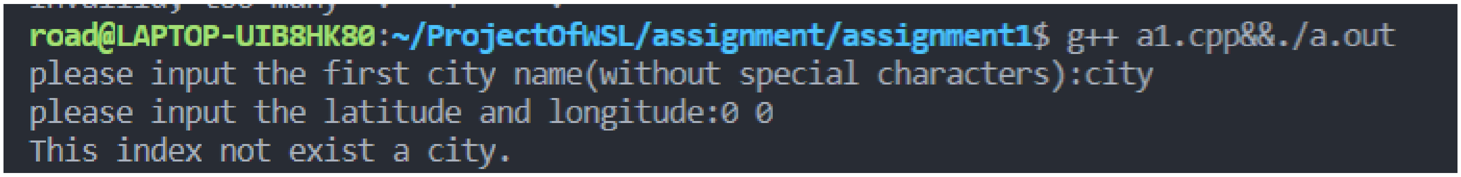
```
input:
city
+-2 4

output:
Invallid, too many '.' '+' '-'.
```



```
road@LAPTOP-UIB8HK80:~/ProjectOfWSL/assignment/assignment1$ g++ a1.cpp&&./a.out
please input the first city name(without special characters):city
please input the latitude and longitude:+-2 4
Invallid, too many '.' '+' '-'.
```

Test case #5: **经纬度为(0,0)时不存在一座城市**

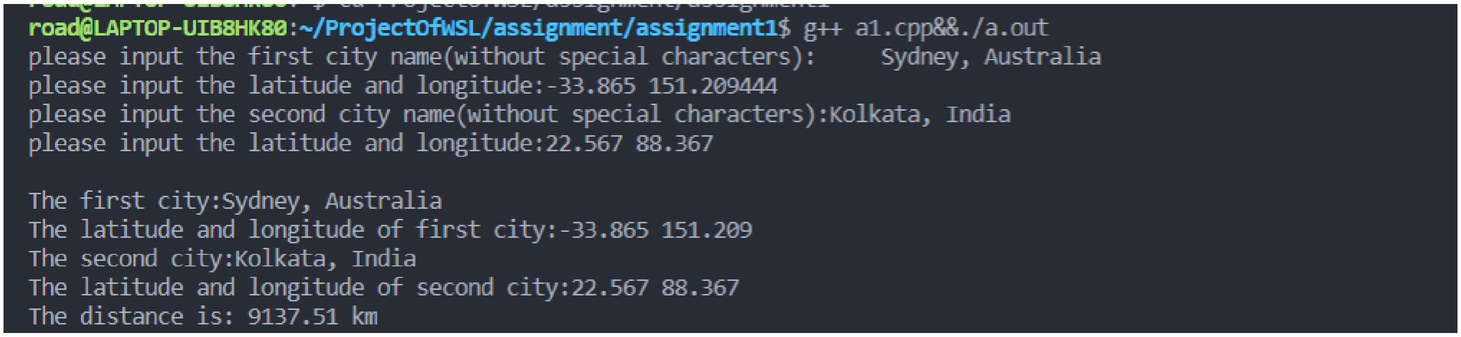
```
input:
city
0 0
output:
This index not exist a city.
```



```
road@LAPTOP-UIB8HK80:~/ProjectOfWSL/assignment/assignment1$ g++ a1.cpp&&./a.out
please input the first city name(without special characters):city
please input the latitude and longitude:0 0
This index not exist a city.
```

Test case #6: **除去城市名字前后的多余空格**

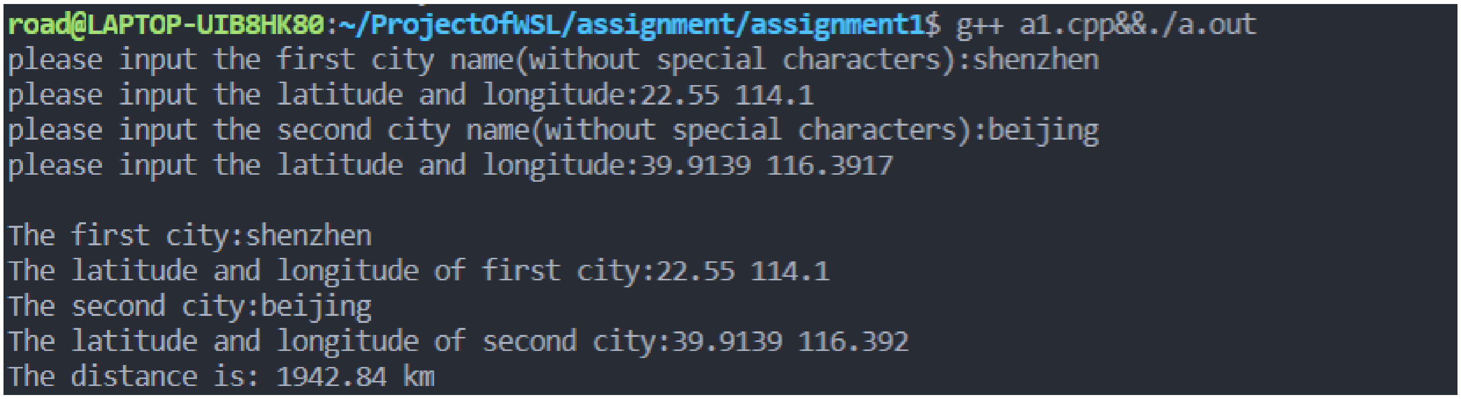
```
input:
     Sydney, Australia
-33.865 151.209444
Kolkata, India
22.567 88.367

output:
The first city:Sydney, Australia
The latitude and longitude of first city:-33.865 151.209
The second city:Kolkata, India
The latitude and longitude of second city:22.567 88.367
The distance is: 9137.51 km
```



```
road@LAPTOP-UIB8HK80:~/ProjectOfWSL/assignment/assignment1$ g++ a1.cpp&&./a.out
please input the first city name(without special characters):     Sydney, Australia
please input the latitude and longitude:-33.865 151.209444
please input the second city name(without special characters):Kolkata, India
please input the latitude and longitude:22.567 88.367

The first city:Sydney, Australia
The latitude and longitude of first city:-33.865 151.209
The second city:Kolkata, India
The latitude and longitude of second city:22.567 88.367
The distance is: 9137.51 km
```

Test case #7: **正常案例**

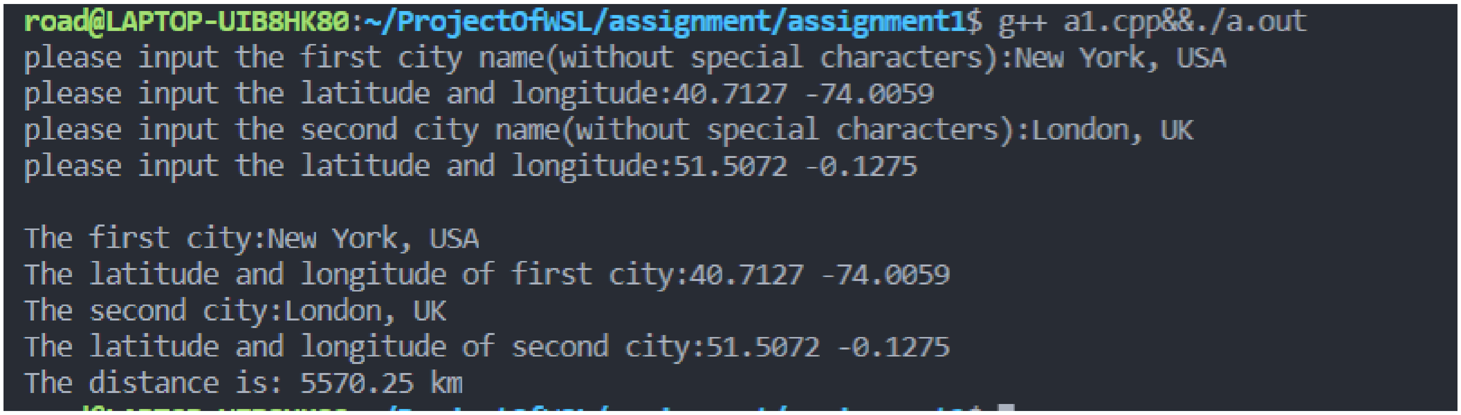
```
input:
Shenzhen
22.55 114.1
Beijing
39.9139 116.3917

output:
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 116.392
The distance is: 1942.84 km
```



```
road@LAPTOP-UIB8HK80:~/ProjectOfWSL/assignment/assignment1$ g++ a1.cpp&&./a.out
please input the first city name(without special characters):shenzhen
please input the latitude and longitude:22.55 114.1
please input the second city name(without special characters):beijing
please input the latitude and longitude:39.9139 116.3917

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 116.392
The distance is: 1942.84 km
```

Test case #8: **正常案例**

```
input:
New York, USA
40.7127 -74.0059
London, UK
51.5072 -0.1275

output:
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 is: 5570.25 km
```



```
road@LAPTOP-UIB8HK80:~/ProjectOfWSL/assignment/assignment1$ g++ a1.cpp&&./a.out
please input the first city name(without special characters):New York, USA
please input the latitude and longitude:40.7127 -74.0059
please input the second city name(without special characters):London, UK
please input the latitude and longitude:51.5072 -0.1275

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 is: 5570.25 km
```

part 4 - Difficulties & Solutions

> 1. 输入的情况比较多，需要对各种情况分类讨论情况,考虑了6种异常情况，对应上述测试案例
> 2. 对于输入经纬度格式不正确问题，一开始设计直接设计成了double,导致对于异常的处理十分困难，后面想到可以化成string先进行格式上的判断处理后，再将其转化为double类型进行处理
> 3. 之前常出现还没完全输入数据，就已经输出了，发现时由于我们输入数据时换行导致的，加上cin.get(); 处理掉缓冲区内的\n即可