**Assignmen 1**

**Name**: 何宜芮(He Yirui)

**SID**: 11811031

**Part 1 – Analysis**

For this problem, we need to get input first. Using “getline(cin,str)” to get the name of cities first. Then, use “cin>>str” to get the longitude and the latitude.

The problem is to calculate the distance between two cities by using the longitude and latitude of two cities.

From the formula for computing distance, we know that:

**phi = 90 – latitude**

and theta:

**theta = longitude**

We denote two cities as city1 and city2, using the formulas above, we know phi1 and theta1 of the first city as well as phi2 and theta2 of city2.

Then we use the formula **phi/360\*2π** and theta**/360\*2π** to get the radians.

Then, we use

**c = sin(phi1) \* sin(phi2) \* cos(theta1-theta2) +cos(phi1) \* cos(phi2)**

and

**d = R\*arccos(c)(R = 6371 km)**

to get the distance of two cities.

Finally, we use “cout<<” to print them.

**Part 2 - Code**

#include **<iostream>**#include **<math.h>  
  
using namespace** std;  
string city1, city2, rub;  
**double** longitude1, latitude1, latitude2, longitude2, phi1, theta1, phi2, theta2, c, d;  
**double** pi = **M\_PI**;  
  
string trim(string);  
  
**int** main() {  
 cout << **"The first city:"**;  
 getline(cin, city1);  
 city1 = trim(city1);  
 **if** (city1.length() == 0) {  
 cout << **"Input is not correct."**;  
 **return** 0;  
 }  
 cout << **"The latitude and longitude of first city:"**;  
 cin >> latitude1 >> longitude1;  
 **if** (!cin.fail()) {  
 **if** (latitude1 < -90 || latitude1 > 90) {  
 cout << **"Input is not correct."**;  
 **return** 0;  
 }  
 **if** (longitude1 < -180 || longitude1 > 180) {  
 cout << **"Input is not correct."**;  
 **return** 0;  
 }  
 getline(cin, rub);  
 rub = trim(rub);  
 **if** (rub.size() != 0) {  
 cout << **"Input is not correct."**;  
 **return** 0;  
 }  
 cout << **"The second city:"**;  
 getline(cin, city2);  
 city2 = trim(city2);  
 **if** (city2.length() == 0) {  
 cout << **"Input is not correct."**;  
 **return** 0;  
 }  
 cout << **"The latitude and longitude of second city:"**;  
 cin >> latitude2 >> longitude2;  
 **if** (!cin.fail()) {  
 getline(cin, rub);  
 rub = trim(rub);  
 **if** (rub.size() != 0) {  
 cout << **"Input is not correct."**;  
 **return** 0;  
 }  
 **if** (latitude2 < -90 || latitude2 > 90) {  
 cout << **"Input is not correct."**;  
 **return** 0;  
 }  
 **if** (longitude2 < -180 || longitude2 > 180) {  
 cout << **"Input is not correct."**;  
 **return** 0;  
 }  
 phi1 = (90 - latitude1) / 180 \* pi;  
 phi2 = (90 - latitude2) / 180 \* pi;  
 theta1 = longitude1 / 180 \* pi;  
 theta2 = longitude2 / 180 \* pi;  
 c = sin(phi1) \* sin(phi2) \* cos(theta1 - theta2) + cos(phi1) \* cos(phi2);  
 d = acos(c) \* 6371;  
 cout << **"The distance between "** << city1 << **" and "** << city2 << **" is "** << d << **" km"** << endl;  
 **return** 0;  
 } **else** {  
 cout << **"Input is not correct."**;  
 **return** 0;  
 };  
 } **else** {  
 cout << **"Input is not correct."**;  
 **return** 0;  
 }  
}  
  
string trim(string st) {  
 **while** (st[0] == **' '**) {  
 st = st.substr(1, st.length() - 1);  
 }  
 **while** (st[st.length() - 1] == **' '**) {  
 st = st.substr(0, st.length() - 2);  
 }  
 **return** st;  
}

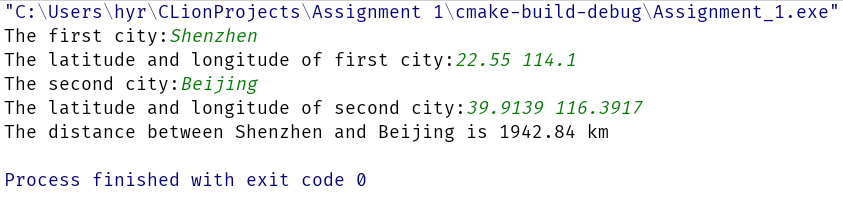
**Part 3 - Result & Verification**

Test case #1:

Input: Shenzhen 22.55 114.1

Beijing 39.9139 116.3917

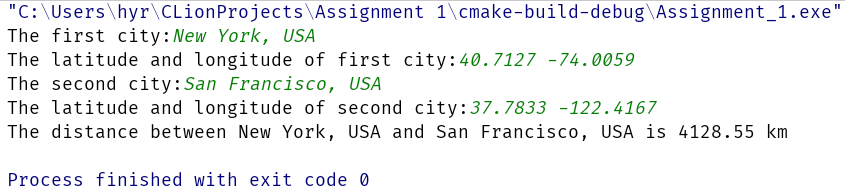
Output: The distance between Shenzhen and Beijing is 1942.84 km



Test case #2:

Input: New York, USA 40.7127 -74.0059

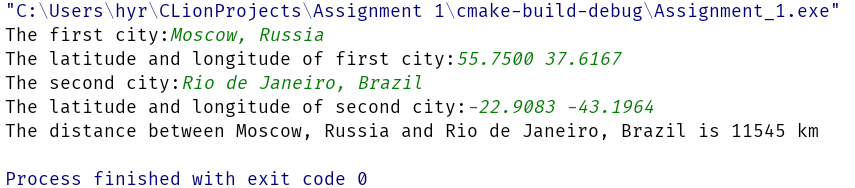
Output: San Francisco, USA 37.7833 -122.4167



Test case #3

Input: Moscow, Russia 55.7500 37.6167

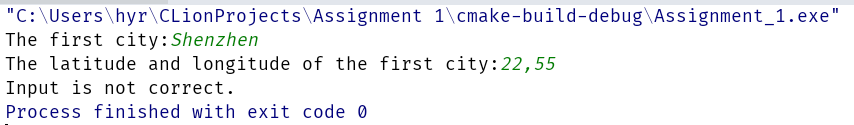
Output: Rio de Janeiro, Brazil -22.9083 -43.1964



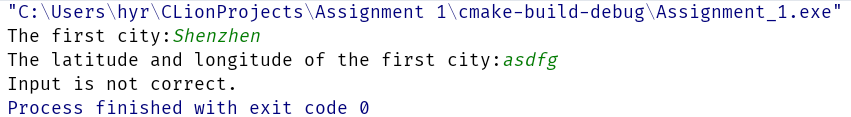
Test case#4

Wrong inputs.

(1)Input numbers are in incorrect form.

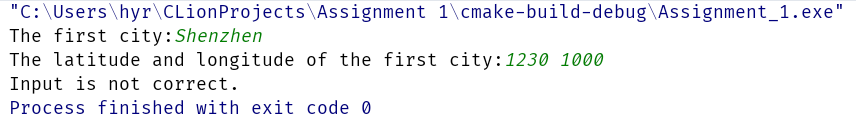


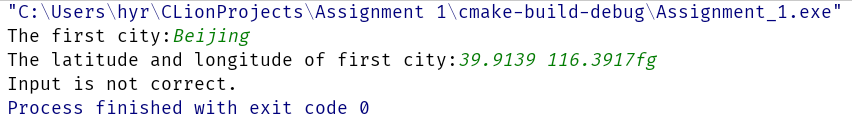
(2)Input incorrect things.



(3)Longitude or latitude is not between the correct ranges.

(-180 <= longitude <= 180 and -90 <= latitude <= 90)





**Part 4 - Difficulties & Solutions**

We can’t get a string with space by using “cin>>str”.Using “getline(cin, str)” to get a string with space inside of it.

We use “cin>>str”and “getline(cin, str)” at the same time, there cased some problems. Because “cin>>str” will not get the “\n” away when the line changed. And “getline(cin, str)” stops scanning when it meet “\n”. Using “getline(cin, str)” after using “cin>>str” can solve this problem.

Also, we need to judge if the input is reasonable, so I use “if…else…” to see if longitude and latitude are numbers and if they are in the correct range.

When someone input an incorrect parameter (like program requires to input a number but user inputs a char ). We can use “if(cin.fail())” to avoid program crashing.

Maybe users will input something like “ (spaces) Shenzhen” in first or third input. I wrote a function called “trim(string str)” to take those spaces away.

Maybe users will input something like “Beijing 39.9139 116.3917ffd” in second and fourth input. I use a string called rub to see whether the input is correct or not. I also，use trim to help to check these input, because inputs like “Beijing 39.9139 116.3917 (spaces) ” seem in the right form.