CS205 C/ C++ Programming - Lab Assignment 3

Name: 邱煜（Qiu Yu）

SID: 11611127

Part1 - Analysis

The problem is to calculate the flying distance between two cities. The cities should in the given file `world_cities.csv`. This csv file stores some cities' `city name`, `province or state name(may be absent)`, `country name`, `latitude`, `longtitude`.

The user just need to enter two cities' name only. No need to input their latitudes and longtitudes. This program will search their latitudes and longtitudes from file `world_cities_csv`.

And then the program use the below function to calculate the distance between the two cities and output the result.

> 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)

The program should be in a loop that keeps running until the user input the exit command `bye` (case-insensitive)

And there're some special situation need to handle:

1. The maximum length for name is 25. (there exists a city whose name is larger than 25)
2. The size of the array which stores all the cities' information is 800. (The total number is 988 rows.)
3. The file `world_cities.csv` isn't found.
4. The input city isn't found or its length < 3.
5. The input is "New York" and "New York City" must be retrieved.
6. The input is "New" and the list of the cities whose name starts of "new" should be displayed.

Part2 - Code

```cpp
//
//  main.cpp
//  lab3
//
//  Created by 邱煜 on 2019/3/29.
//  Copyright © 2019 邱煜. All rights reserved.
//

#include <iostream>
#include <fstream>
#include <string>
#include <cmath>
#include <cstdlib>
#include <algorithm>
#include "def.hpp"

using namespace std;

City cities[arraySize];
int citisNum;
string fCity, sCity;

int main() {
    double fLati, fLong, sLati, sLong;
    double result;

    ifstream file ("world_cities.csv");
    if(!file.is_open()){
        cout << "Open file failed!" << endl;
        exit(9);
    }
    string value;
    int i = 0;
    while(file.good()){
        // city name
        getline(file, value, ',');
        if(value.length())
        {
            cities[i].cityName = value;
            if(value.length()>nameMaxLen){
                cities[i].cityName = value.substr(0,nameMaxLen);
                cout << value << " is truncated to " <<
value.substr(0,nameMaxLen) << " because its name is too long." << endl;
                cout << "So please enter the name after truncated if you
want to enter this city." << endl;
            }
        }
        // province name(ignored)
```

```
        getline(file, value, ',');
        // country name
        getline(file, value, ',');
        if(value.length())
        {
            cities[i].countryName = value;
        }
        // latitude
        getline(file, value, ',');
        if(value.length())
        {
            cities[i].latitude = atof(value.c_str());
        }
        // longtitude
        getline(file, value);
        if(value.length())
        {
            cities[i].longtitude = atof(value.c_str());
        }
        // handle the situation that the number of city > the array size
        if(i>arraySize-2){
            cout << "The cities has been filled, others data after " << i+1 << " row are not loaded." << endl;
            break;
        }
        i++;
    }
    file.close();
    citisNum = i-1;
    if(i<arraySize-1){
        cout << "Loading successfully!" << endl;
    }
    //verification
    // cout << cities[598].latitude << ", "<< cities[598].longtitude << endl;
//    for (int j = 0; j<i-1; j++) {
//        cout << j << cities[j].cityName << ", "<< cities[j].countryName;
//        cout << cities[j].latitude << cities[j].longtitude << endl;
//    }

    while(1){
        cout << "Enter the first city's name(Enter \"bye\" to exit): ";
        getline(cin, fCity);
        trim(fCity);
        if(toUpper(fCity) == "BYE"){
            cout << "Bye" << endl;
            exit(0);
        }
```

```
93          int idx1 = handleInput(fCity, 1);
94          while(idx1 == -1){
95              cout << "Enter the first city's name(Enter \"bye\" to exit):
";
96              getline(cin, fCity);
97              trim(fCity);
98              if(toUpper(fCity) == "BYE"){
99                  cout << "Bye" << endl;
100                 exit(0);
101             }
102             idx1 = handleInput(fCity, 1);
103         }
104
105         cout << "Enter the second city's name(Enter \"bye\" to exit): ";
106         getline(cin, sCity);
107         trim(sCity);
108         if(toUpper(sCity) == "BYE"){
109             cout << "Bye" << endl;
110             exit(0);
111         }
112         int idx2 = handleInput(sCity,2);
113         while(idx2 == -1){
114             cout << "Enter the second city's name(Enter \"bye\" to exit):
";
115             getline(cin, sCity);
116             trim(sCity);
117             if(toUpper(sCity) == "BYE"){
118                 cout << "Bye" << endl;
119                 exit(0);
120             }
121             idx2 = handleInput(sCity,2);
122         }
123
124         fLati = cities[idx1].latitude;
125         fLong = cities[idx1].longtitude;
126         sLati = cities[idx2].latitude;
127         sLong = cities[idx2].longtitude;
128         result = calculate(fLati, fLong, sLati, sLong);
129         cout << "The distance between " << fCity <<  " and " << sCity <<
" is " << result << " km." << endl;
130     }
131     return 0;
132 }
133
134 // 0 <= phi <= 180 -180 <= theta <= 180
135 // -90 <= latitude <= 90 -180 <= longtitude <= 180
136 double calculate(double fLati, double fLong, double sLati, double sLong){
137     double phi1, phi2, theta1, theta2;
138     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;
}

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

// handle all cases - return the city's index in the cities array
// fOrS - 1 means this is the first city; 2 means second.
int handleInput(string input, int fOrS){
    // Invalid case - #characters < 3
    if(input.length() < 3){
        cout << "Invalid input - Less than 3 characters." << endl;
        return -1;
    }

    // "New" case
    if(toUpper(input) == "NEW"){
        string choice = "0";
        cout << "There're several cities whose name starts with \"New\", please select the correct one by the index." << endl;
        cout << "1: New Delhi\n2: New Orleans\n3: New York City\n4: Newcastle upon Tyne\n5: Newcastle" << endl;
        getline(cin, choice);
        trim(choice);
        char *end;
        while(choice!="1" && choice!="2" && choice!="3" && choice!="4" && choice!="5") {
            cout << "Invalid choice" << endl;
            cout << "1: New Delhi\n2: New Orleans\n3: New York City\n4: Newcastle upon Tyne\n5: Newcastle" << endl;
            getline(cin, choice);
            trim(choice);
        }
        if(choice=="1"){
            if(fOrS == 1)fCity = "New Delhi";
            else if(fOrS == 2)sCity = "New Delhi";
        }else if(choice=="2"){
            if(fOrS == 1)fCity = "New Orleans";
            else if(fOrS == 2)sCity = "New Orleans";
```

```
        }else if(choice=="3"){
            if(fOrS == 1)fCity = "New York City";
            else if(fOrS == 2)sCity = "New York City";
        }else if(choice=="4"){
            if(fOrS == 1)fCity = "Newcastle upon Tyne";
            else if(fOrS == 2)sCity = "Newcastle upon Tyne";
        }else if(choice=="5"){
            if(fOrS == 1)fCity = "Newcastle";
            else if(fOrS == 2)sCity = "Newcastle";
        }
        return (int)strtol(choice.c_str(), &end, 10) + 595;
    }

    // "New York" case
    if (toUpper(input) == "NEW YORK") {
        if(fOrS == 1)fCity = "New York City";
        else if(fOrS == 2)sCity = "New York City";
        return 598;
    }

    // other valid cases
    return findIndex(input, fOrS);

}

// handle valid cases - return the index of the city in the cities array
int findIndex(string cityName, int fOrS){
    for (int i = 0; i<citisNum; i++) {
        if (toUpper(cities[i].cityName) == toUpper(cityName)) {
            return i;
        }
    }
    cout << "City " << cityName << " not found!" << endl;
    return -1; // not found
}

// clear the whitespaces on both ends of the input string
void trim(string &str){
    if(!str.empty()){
        str.erase(0,str.find_first_not_of(" "));
        str.erase(str.find_last_not_of(" ") + 1);
    }
}

// to upper the input string
string toUpper(string str){
    string new_str = str;
    transform(new_str.begin(), new_str.end(), new_str.begin(), (int (*)
(int))toupper);
```

```
    return new_str;
}
```

```
//
//  def.hpp
//  lab3
//
//  Created by 邱煜 on 2019/3/29.
//  Copyright © 2019 邱煜. All rights reserved.
//

#ifndef def_hpp
#define def_hpp
#include <string>
#define Pi 3.141592654
#define R 6371

const int arraySize = 1000;
const int nameMaxLen = 35;

double calculate(double fLati, double fLong, double sLati, double sLong);
double convert(double angle);
int handleInput(std::string input, int fOrS);
int findIndex(std::string input, int fOrS);
void trim(std::string &str);
std::string toUpper(std::string str);

struct City{
    std::string cityName;
    std::string countryName;
    double latitude;
    double longtitude;
};

#endif /* stru_h */
```

Part 3 - Result & Verification

Requirements part1

1.

```
 Personal@Coreys  ~/github/C-CPP-Assignment/lab3  master ●  ./main
Las Palmas de Gran Canaria is truncated to Las Palmas de Gran Canari because its
 name is too long.
So please enter the name after truncated if you want to enter this city.
The cities has been filled, others data after 800 row are not loaded.
Enter the first city's name(Enter "bye" to exit): bye
Bye
 Personal@Coreys  ~/github/C-CPP-Assignment/lab3  master ●
```

2.

```
 Personal@Coreys  ~/github/C-CPP-Assignment/lab3  master ●  ./main
Loading successfully!
Enter the first city's name(Enter "bye" to exit): bye
Bye
 Personal@Coreys  ~/github/C-CPP-Assignment/lab3  master ●
```

3.

```
 Personal@Coreys  ~/github/C-CPP-Assignment/lab3  master ●  ./main
Open file failed!
 ✘ Personal@Coreys  ~/github/C-CPP-Assignment/lab3  master ●
```

4.

```
 Personal@Coreys  ~/github/C-CPP-Assignment/lab3  master ●  ./main
Loading successfully!
Enter the first city's name(Enter "bye" to exit): bye
Bye
 Personal@Coreys  ~/github/C-CPP-Assignment/lab3  master ●
```

Part 2

1.

Case - bye

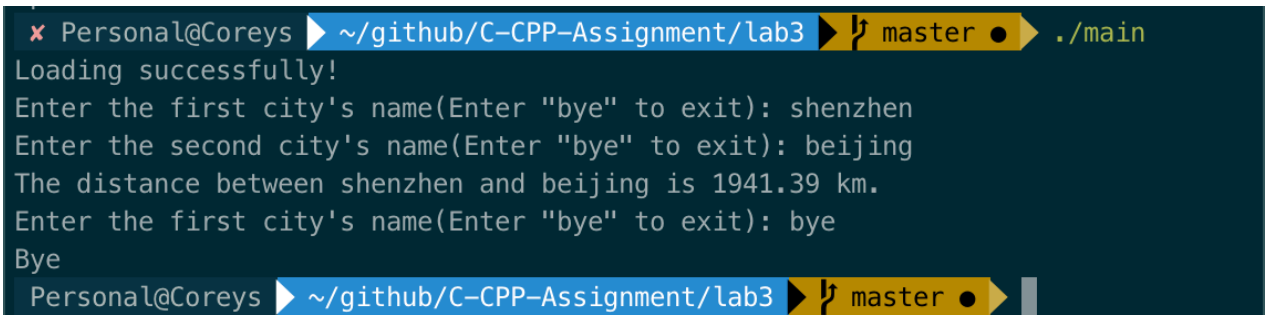
```
 Personal@Coreys  ~/github/C-CPP-Assignment/lab3  master ●  ./main
Loading successfully!
Enter the first city's name(Enter "bye" to exit): bye
Bye
 Personal@Coreys  ~/github/C-CPP-Assignment/lab3  master ●
```

```
 Personal@Coreys  ~/github/C-CPP-Assignment/lab3  master ●  ./main
Loading successfully!
Enter the first city's name(Enter "bye" to exit): ByE
Bye
 Personal@Coreys  ~/github/C-CPP-Assignment/lab3  master ●
```

Normal case

Test case #1:

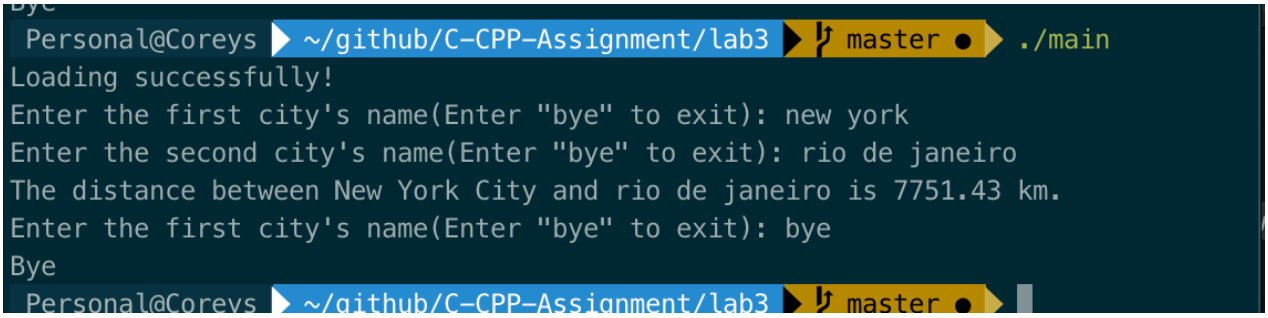
```
Input:
    shenzhen
    beijing
Output:
    1941.39
Verification:
    1950
```



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:

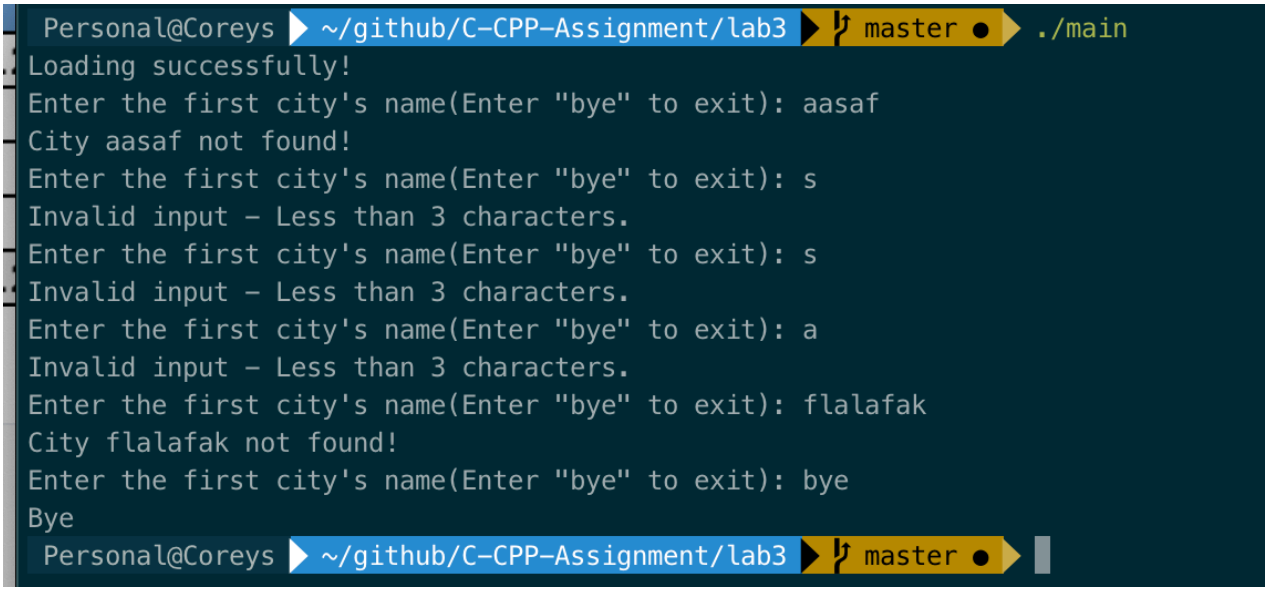
```
Input:
    new york
    rio de janeiro
Output:
    7751
Verification:
    7760
```



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.

Case - not found & length < 3

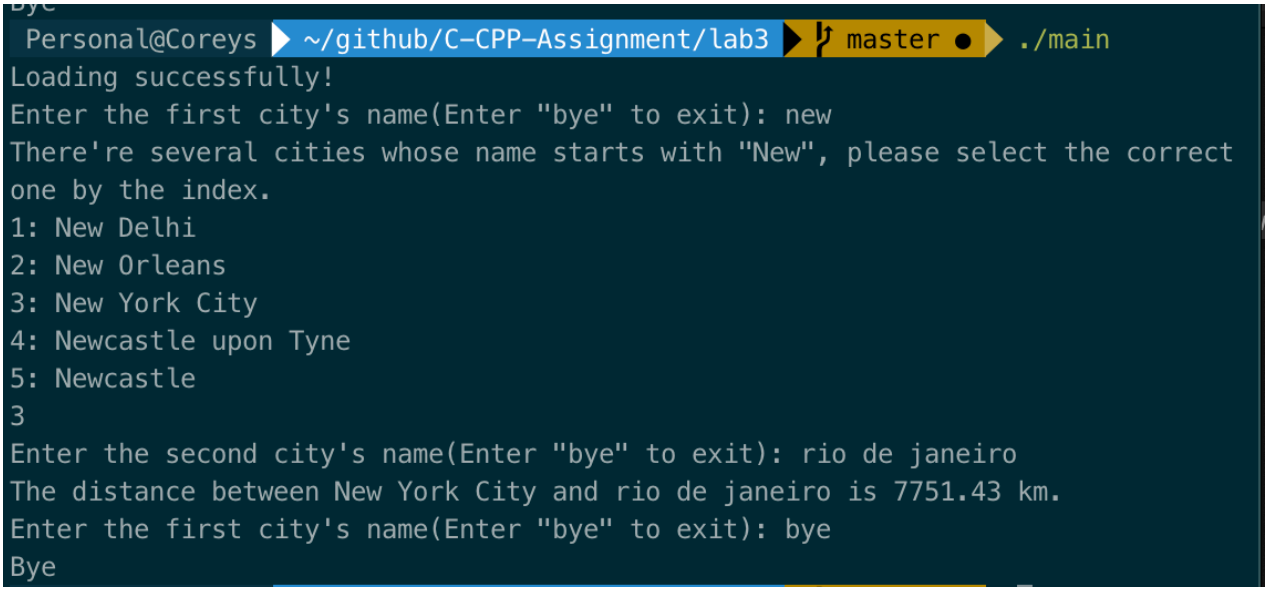


3.

Case - "New York" and "New"

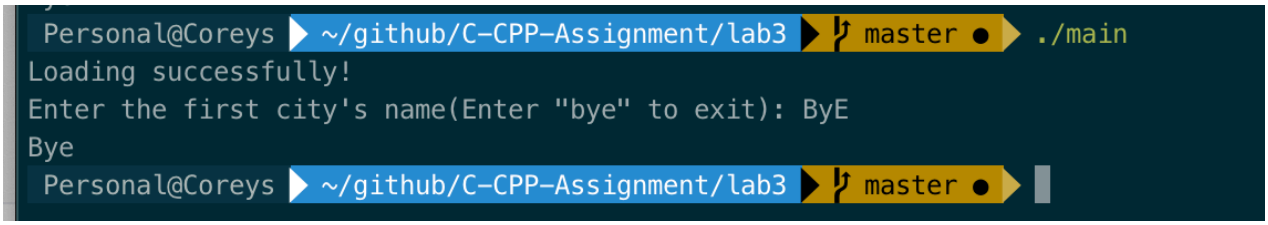
Test case:

```
Input:
    new
    rio de janeiro
    3 (means New York City)
Output:
    7751
Verification:
    7760
```



4.

Case - ignore the whitespaces on both ends.



Part 4 - Difficulties & Solutions

1. Some function needs to implements. For example, `ToUpper ()` , `trim ()` . Then I implements them myself.
2. C++ could detect the overflow of the array. I assign a variable to record how much line it have read. Then once it larger than the pre-setting array size, the program stop reading file and display it.