

# Assignment – 3

---

**Name:** 何宜芮(He Yirui)

**SID:** 11811031

## Part 1 – Analysis

---

The part 1 requires us to read a “.csv” file and to load the information about the cities. First, noticing that all constant values must be set as #define so that they can be easily changed. Then, we are required to create an array containing structures as I have defined them, read the file, and load their names, country names, latitudes and longitudes into the array.

Initially set the maximum length for names (city and country name) to 25, and the array size to 800. When array is full, I should print a warning and stop reading the remaining file.

Then set the maximum length for names to 35, and the array size to 1000. Ensure that the program is reading without problems.

Rename the file to world\_cities.tmp. Run the program. It mustn't crash and should display a warning about the missing file.

As for part 2, after all data is loaded, the program should enter a loop from which it will exit when the user types "bye" (case-insensitive).

In each circle, it will prompt for the names of two cities. The program must search the cities in the table by their names and retrieve the latitudes and longitudes. If the city isn't found or if the length of the name is shorter than three letters, a message must be displayed and the user must be prompted for another name. The city names should be case-insensitive. After matching the input and names on the list, the list of the matched cities must be displayed, and ask the user prompted for the right one.

Finally, the names of the cities (as stored in the memory) and the distance between them must be displayed. As before, your program should ignore all whitespaces on both ends of user inputs.

## Part 2 – Code

---

```
#include <iostream>
```

```
#include <string>
```

```
#include <fstream>
```

```
#include <sstream>
```

```
#include <algorithm>
```

```
#include <vector>
```

```
#include <cmath>
```

```
#define arSize 1000
```

```
#define nameLength 35
```

```
using namespace std;
```

```
string trim(string);
```

```
bool exit(string);
```

```
void toLower(string *str);
```

```
bool is_number(const std::string &s);
```

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

```
int main() {  
    city cityAr1[arSize];  
    string str;  
    ifstream fin("world_cities.csv");  
    if (!fin.good()) {  
        cout << "Fail to open the file."  
        return 0;  
    }  
    int count = 0;  
    while (getline(fin, str)) {  
        if (count >= arSize) {  
            cout << "Input fail since the array is full!\n";  
            break;  
        }  
        istringstream sin(str);  
        string name, country, latitude, longitude;  
        getline(sin, name, ',');  
        getline(sin, country, ','); //This is to deal with the name of state .  
        getline(sin, country, ',');  
        getline(sin, latitude, ',');  
        getline(sin, longitude);  
        sin.clear();  
        if (name.length() > nameLength || country.length() > nameLength) {  
            cout << "The length of names should not bigger than " << nameLength << ". This data is not  
loaded\n";  
            continue;  
        }  
        cityAr1[count].name = trim(name);
```

```

        cityAr1[count].country = trim(country);
        cityAr1[count].latitude = stod(trim(latitude));
        cityAr1[count].longitude = stod(trim(longitude));
        count++;
    }
    fin.close();

    //Part two;
    city city1;
    city city2;
    city cityAr2[arSize];
    copy(begin(cityAr1), end(cityAr1), begin(cityAr2));
    for (int i = 0; i < count; i++) {
        toLower(&(cityAr2[i].country));
        toLower(&(cityAr2[i].name));
    }
    bool loop = true;
    vector<city> list;

    while (loop) {
        list.clear();
        while (true) {
            cout << "Please input the first city: ";
            getline(cin, str);
            str = trim(str);
            toLower(&str);
            if (exit(str)) {
                loop = false;
                break;
            }
            if (str.length() < 3) {
                cout << "Please enter a name more than 3 letters.\n";
                continue;
            } else {
                for (int i = 0; i < count; i++) {
                    if (cityAr2[i].name.find(str) != string::npos) {
                        list.push_back(cityAr1[i]);
                    }
                }
            }
            if (list.empty()) {
                cout << "Please input another name which is on the list.\n ";
                continue;
            } else {

```

```

cout << "Please choose the right city:(please input the number!) \n";
for (int i = 0; i < list.size(); i++) {
    cout << i + 1 << ". " << list[i].name << ", " << list[i].country << ", latitude: "
        << list[i].latitude
        << ", longitude: " << list[i].longitude << "\n";
}
int num;
getline(cin,str);
str = trim(str);
toLower(&str);
if (exit(str)) {
    loop = false;
    break;
}
while (true) {
    if (is_number(str)) {
        num = stoi(str);
        if (num <= list.size() && num >= 1)break;
        else{
            cout << "Please input the right number: ";
            getline(cin,str);
            str = trim(str);
            toLower(&str);
            if (exit(str)) {
                loop = false;
                break;
            }
        }
    }
    else {
        cout << "Please input the right number: ";
        getline(cin,str);
        str = trim(str);
        toLower(&str);
        if (exit(str)) {
            loop = false;
            break;
        }
    }
}
if(!loop)break;
cout << "You have successfully chosen your first city.\n";
city1.name = list[num - 1].name;
city1.country = list[num - 1].country;
city1.longitude = list[num - 1].longitude;

```

```

        city1.latitude = list[num - 1].latitude;
        cout << city1.name << ", " << city1.country << ", latitude: "
            << city1.latitude << ", longitude: " << city1.longitude << "\n";
        break;
    }
}
if (!loop)break;
list.clear();
while (true) {
    cout << "Please input the second city: ";
    getline(cin, str);
    str = trim(str);
    toLower(&str);
    if (exit(str)) {
        loop = false;
        break;
    }
    if (str.length() < 3) {
        cout << "Please enter a name more than 3 letters.\n";
        continue;
    } else {
        for (int i = 0; i < count; i++) {
            if (cityAr2[i].name.find(str) != string::npos) {
                list.push_back(cityAr1[i]);
            }
        }
    }
    if (list.empty()) {
        cout << "Please input another name which is on the list.\n";
        continue;
    } else {
        cout << "Please choose the right city:(please input the number!) \n";
        for (int i = 0; i < list.size(); i++) {
            cout << i + 1 << ". " << list[i].name << ", " << list[i].country << ", latitude: "
                << list[i].latitude
                << ", longitude: " << list[i].longitude << "\n";
        }
        int num;
        getline(cin, str);
        str = trim(str);
        toLower(&str);
        if (exit(str)) {
            loop = false;
            break;
        }
    }
}

```

```

    }
    while (true) {
        if (is_number(str)) {
            num = stoi(str);
            if (num <= list.size() && num >= 1)break;
            else{
                cout << "Please input the right number: ";
                getline(cin,str);
                str = trim(str);
                toLower(&str);
                if (exit(str)) {
                    loop = false;
                    break;
                }
            }
        } else {
            cout << "Please input the right number: ";
            getline(cin,str);
            str = trim(str);
            toLower(&str);
            if (exit(str)) {
                loop = false;
                break;
            }
        }
    }
    if(!loop)break;
    cout << "You have successfully chosen your second city.\n";

    city2.name = list[num - 1].name;
    city2.country = list[num - 1].country;
    city2.longitude = list[num - 1].longitude;
    city2.latitude = list[num - 1].latitude;
    cout << city2.name << ", " << city2.country << ", latitude: "
        << city2.latitude << ", longitude: " << city2.longitude << "\n";
    break;
}
}

if (!loop)break;
double phi1, phi2, theta1, theta2, d, c;
const double pi = 3.1415926535;

phi1 = (90 - city1.latitude) / 180 * pi;
phi2 = (90 - city2.latitude) / 180 * pi;

```

```

        theta1 = city1.longitude / 180 * pi;
        theta2 = city2.longitude / 180 * pi;
        c = sin(phi1) * sin(phi2) * cos(theta1 - theta2) + cos(phi1) * cos(phi2);
        d = acos(c) * 6371;
        cout << "The distance between " << city1.name << "(" << city1.country << ")" << " and " <<
city2.name << "("
                << city2.country << ")" << " is " << d << " km" << endl;
    }
    return 0;
}

```

```

string trim(string st) {
    while (isspace(st[0])) {
        st = st.substr(1, st.length() - 1);
    }
    while (isspace(st[st.length() - 1])) {
        st = st.substr(0, st.length() - 1);
    }
    return st;
}

```

```

bool exit(string str) {
    transform(str.begin(), str.end(), str.begin(), ::tolower);
    if (str.compare("bye") == 0) return true;
    else return false;
}

```

```

void toLower(string *str) {
    transform((*str).begin(), (*str).end(), (*str).begin(), ::tolower);
}

```

```

bool is_number(const std::string &s) {
    std::string::const_iterator it = s.begin();
    while (it != s.end() && std::isdigit(*it)) ++it;
    return !s.empty() && it == s.end();
}

```

*//This function is from*

*stackoverflow :<https://stackoverflow.com/questions/4654636/how-to-determine-if-a-string-is-a-number-with-c>*

## Part 3 - Result & Verification

---

### Test case #1 :

Set the maximum length for names (city and country name) to 25, and the array size to 800.

```
C:\Users\hyr\CLionProjects\Assignment_3\cmake-build-debug\Assignment_3.exe
The length of names should not bigger than 25. This data is not loaded
The length of names should not bigger than 25. This data is not loaded
The length of names should not bigger than 25. This data is not loaded
The length of names should not bigger than 25. This data is not loaded
The length of names should not bigger than 25. This data is not loaded
Input fail since the array is full!
Please input the first city:
```

(When the city and country name is more than 25)

## Test case #2 :

Set the maximum length for names to 35, and the array size to 1000. Check that your program runs without any problem.

```
C:\Users\hyr\CLionProjects\Assignment_3\cmake-build-debug\Assignment_3.exe
Please input the first city: |
```

(This means the program went to the second part without any mistakes.)

## Test case #3 :

Rename the file to world\_cities.tmp. Run your program. It mustn't crash and should display a warning about the missing file.

```
C:\Users\hyr\CLionProjects\Assignment_3\cmake-build-debug\Assignment_3.exe
Fail to open the file.
Process finished with exit code 0
```

## Test case #4:

Common way to search the cities.

```
C:\Users\hyr\CLionProjects\Assignment_3\cmake-build-debug\Assignment_3.exe
Please input the first city: Beijing
Please choose the right city:(please input the number!)
1. Beijing, China, latitude: 39.9, longitude: 116.4
j
You have successfully chosen your first city.
Beijing, China, latitude: 39.9, longitude: 116.4
Please input the second city: New York
Please choose the right city:(please input the number!)
1. New York City, United States, latitude: 40.667, longitude: -73.933
j
You have successfully chosen your second city.
New York City, United States, latitude: 40.667, longitude: -73.933
The distance between Beijing(China) and New York City(United States) is 10995.5 km
Please input the first city: bye
Process finished with exit code 0
```

## Test case #5 :

Show that the input is case-insensitive.

```
C:\Users\hyr\CLionProjects\Assignment_3\cmake-build-debug\Assignment_3.exe
Please input the first city: york
Please choose the right city:(please input the number!)
1. New York City, United States, latitude: 40.667, longitude: -73.933
2. York, United Kingdom, latitude: 53.95, longitude: -1.067
j
You have successfully chosen your first city.
New York City, United States, latitude: 40.667, longitude: -73.933
Please input the second city: bei
Please choose the right city:(please input the number!)
1. Beijing, China, latitude: 39.9, longitude: 116.4
2. Beirut, Lebanon, latitude: 33.883, longitude: 35.517
3. Ribeirão Preto, Brazil, latitude: -21.183, longitude: -47.8
4. Taipei, China, latitude: 25.033, longitude: 121.633
j
You have successfully chosen your second city.
Beijing, China, latitude: 39.9, longitude: 116.4
The distance between New York City(United States) and Beijing(China) is 10995.5 km
Please input the first city: BYE
Process finished with exit code 0
```



## Test case #6 :

When the input is less than 3 letters.

```
C:\Users\hyr\CLionProjects\Assignment_3\cmake-build-debug\Assignment_3.exe
Please input the first city: ne
Please enter a name more than 3 letters.
Please input the first city: new
Please choose the right city:(please input the number!)
1. New Delhi, India, latitude: 28.617, longitude: 77.217
2. New Orleans, United States, latitude: 29.967, longitude: -90.05
3. New York City, United States, latitude: 40.667, longitude: -73.933
4. Newcastle upon Tyne, United Kingdom, latitude: 54.967, longitude: -1.617
5. Newcastle, Australia, latitude: -32.917, longitude: 151.75
j
You have successfully chosen your first city.
New York City, United States, latitude: 40.667, longitude: -73.933
Please input the second city: b
Please enter a name more than 3 letters.
Please input the second city: Bei
Please choose the right city:(please input the number!)
1. Beijing, China, latitude: 39.9, longitude: 116.4
2. Beirut, Lebanon, latitude: 33.883, longitude: 35.517
3. Ribeirão Preto, Brazil, latitude: -21.183, longitude: -47.8
4. Taipei, China, latitude: 25.033, longitude: 121.633
j
You have successfully chosen your second city.
Beijing, China, latitude: 39.9, longitude: 116.4
The distance between New York City(United States) and Beijing(China) is 10995.5 km
Please input the first city: Bye

Process finished with exit code 0
```

## Test case #7 :

When choosing the city from the list, input a number which is not on the list.

```
C:\Users\hyr\CLionProjects\Assignment_3\cmake-build-debug\Assignment_3.exe
Please input the first city: new
Please choose the right city:(please input the number!)
1. New Delhi, India, latitude: 28.617, longitude: 77.217
2. New Orleans, United States, latitude: 29.967, longitude: -90.05
3. New York City, United States, latitude: 40.667, longitude: -73.933
4. Newcastle upon Tyne, United Kingdom, latitude: 54.967, longitude: -1.617
5. Newcastle, Australia, latitude: -32.917, longitude: 151.75
0
Please input the right number: 3
You have successfully chosen your first city.
New York City, United States, latitude: 40.667, longitude: -73.933
Please input the second city: bei
Please choose the right city:(please input the number!)
1. Beijing, China, latitude: 39.9, longitude: 116.4
2. Beirut, Lebanon, latitude: 33.883, longitude: 35.517
3. Ribeirão Preto, Brazil, latitude: -21.183, longitude: -47.8
4. Taipei, China, latitude: 25.033, longitude: 121.633
8
Please input the right number: 7
Please input the right number: 4
You have successfully chosen your second city.
Taipei, China, latitude: 25.033, longitude: 121.633
The distance between New York City(United States) and Taipei(China) is 12534.4 km
Please input the first city: bye

Process finished with exit code 0
```

## Test case #8:

Input a name does not match any names on the list.

```
C:\Users\hyr\CLionProjects\Assignment_3\cmake-build-debug\Assignment_3.exe
Please input the first city: nwyuu
Please input another name which is on the list.
Please input the first city: new
Please choose the right city:(please input the number!)
1. New Delhi, India, latitude: 28.617, longitude: 77.217
2. New Orleans, United States, latitude: 29.967, longitude: -90.05
3. New York City, United States, latitude: 40.667, longitude: -73.933
4. Newcastle upon Tyne, United Kingdom, latitude: 54.967, longitude: -1.617
5. Newcastle, Australia, latitude: -32.917, longitude: 151.75
j
You have successfully chosen your first city.
New Delhi, India, latitude: 28.617, longitude: 77.217
Please input the second city: bye

Process finished with exit code 0
```

## Test case #9:

Some invalid inputs.

```

C:\Users\hyr\CLionProjects\Assignment_3\cmake-build-debug\Assignment_3.exe
Please input the first city: new
Please choose the right city:(please input the number!)
1. New Delhi, India, latitude: 28.617, longitude: 77.217
2. New Orleans, United States, latitude: 29.967, longitude: -90.05
3. New York City, United States, latitude: 40.667, longitude: -73.933
4. Newcastle upon Tyne, United Kingdom, latitude: 54.967, longitude: -1.617
5. Newcastle, Australia, latitude: -32.917, longitude: 151.75
hhh
Please input the right number: 1h
Please input the right number: h1
Please input the right number: bye

Process finished with exit code 0

```

## Test case #10:

Test with strange characters.

```

C:\Users\hyr\CLionProjects\Assignment_3\cmake-build-debug\Assignment_3.exe
Please input the first city: Ürümqi
Please choose the right city:(please input the number!)
1. Ürümqi, China, latitude: 43.833, longitude: 87.6
1
You have successfully chosen your first city.
Ürümqi, China, latitude: 43.833, longitude: 87.6
Please input the second city: İzmir
Please choose the right city:(please input the number!)
1. İzmir, Turkey, latitude: 38.433, longitude: 27.15
1
You have successfully chosen your second city.
İzmir, Turkey, latitude: 38.433, longitude: 27.15
The distance between Ürümqi(China) and İzmir(Turkey) is 4985.1 km
Please input the first city: bye

Process finished with exit code 0

```

## Part 4 - Difficulties & Solutions

1. First, I did not know where to place the ".csv" file, so I can open the file. So I used the program to write a file and then found where the file was. That was the default place where the program read the file.
2. When comparing the string, I did not know how to find if a string is a substring of other strings, so I searched on the internet and found ".find()" method.
3. When I am trying to make the list, I try to find a array whose size can change. So I learned to use vector.
4. In order to prevent the program crushing when users input invalid inputs, also when they type "Bye", the program should exit, I tried many ways. Finally I searched a function together with many "while" , "if...else..." to solve the problem.