CS205 C/ C++ Programming - Lab Assignment Template

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

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. 题目通过文件的形式完成输入，我们需要读取文件，将文件的数据存在city数据结构上
> 3. 文件数据读入使用ifstream,在循环中读取所有数据,并将读取到的每行数据以逗号分隔,并初始化为city结构存入数组中，使用bye结束进程
> 4. 在读取的过程中,进行数据的处理与判断,比如城市名应该输入至少三个字母，将城市名前后的空格用trim()去点，模糊匹配时用数字来选取城市
> 5. 计算距离所需要的公式，在作业assignment1中已给出，将其变更对应上即可findDistance(city city1, city city2)

Part2 - Code

```
#include <iostream>
#include <fstream>
#include <cstring>
#include <cmath>
#include <cstdlib> //support for exit()
#include <algorithm>
#include <vector>

using namespace std;

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

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

#define MAX_NAME_LENGTH 35
#define MAX_ARRAY_LENTH 1000
#define FILE_NAME "world_cities.csv"

vector<const city *> choose_City{};

bool end_judge(string str);
string trim(string str);
bool is_number(const std::string &s);
vector<city> readFile(const string &filename);
bool InputCity(const city **p, const vector<city> &cityList);
double findDistance(const city &city1, const city &city2);
void process(vector<city> cityList);

int main(){
    vector<city> cityList = readFile(FILE_NAME);
    if(cityList.size()==0){
       return 0;
    }
    process(cityList);
    cout<<"Progarm is end!"<<endl;
    return 0;
}
void process(vector<city> cityList){
     while (true) {
        const city *c1 = nullptr, *c2 = nullptr;
        while (c1 == nullptr) {
            cout<<"\n-------------------------------------"<<endl;
            cout<<"Input \"bye\" to end the programmer! You need at least input three char"<<endl;
```

```
            cout << "please input the first city name: ";
            if (!InputCity(&c1, cityList)) {
                return ;
            }
        }
        while (c2 == nullptr) {
            cout << "please input the second city name: ";
            if (!InputCity(&c2, cityList)) {
                return ;
            }
        }
        double distance = findDistance(*c1, *c2);
        cout << "The distance between " << c1->name << " and " << c2->name << " is " << distance << " km." << enc
    }
    return;
}
bool end_judge(string str) {
    if (str.size() == 3
        && ('b' == str[0] || 'B' == str[0])
        && ('y' == str[1] || 'Y' == str[1])
        && ('e' == str[2] || 'E' == str[2])
        ) {
        return true;
    }
    return false;
}
string trim(string str) {
    if (str.empty()) return str;
    str.erase(0, str.find_first_not_of(" "));
    str.erase(str.find_last_not_of(" ") + 1);
    return str;
}
bool is_number(const string &str) {
    bool a,b;
    a= !str.empty();
    b= find_if(str.begin(), str.end(),[](unsigned char c) { return !isdigit(c); }) == str.end();
    return a&&b;
}
vector<city> readFile(const string &filename){
    vector<city> cityList;
    ifstream inFile;
    inFile.open(filename);
    if(!inFile.good()){
        cout << "we can\'t open this file: " <<filename <<endl;
        exit(EXIT_FAILURE);
        return cityList;
    }

    int line_serial =1;
    string data;
    while(getline(inFile,data)){
```

```
        if(cityList.size() >= MAX_ARRAY_LENTH){
            cout <<"This file too larger! It more than " << MAX_ARRAY_LENTH <<" lines, operate skipping instructi
        }
        string separate = ",";
        string information[5];
        int num =0;
        size_t index=0;
        //divide
        while((index = data.find(separate)) != string::npos){
            information[num] = data.substr(0,index);
            data.erase(0,index + 1);
            num++;
        }
        information[num] = data;
        if(information[0].size() > MAX_NAME_LENGTH || information[2].size()>MAX_NAME_LENGTH ){
            cout <<"No." <<line_serial<<" name length is larger than "<<MAX_NAME_LENGTH << " skip this line."<<er
        }
        //store
        transform(information[0].begin(),information[0].end(),information[0].begin(),::toupper);
        city city1;
        city1.name = information[0];
        city1.country = information[2];
        city1.latitude = stod(information[3]);
        city1.longitude = stod(information[4]);

        cityList.push_back(city1);
        line_serial++;
    }
    return cityList;

}
bool InputCity(const city **pt, const vector<city> &cityList) {
    string str;
    getline(cin, str);
    str = trim(str);
     if(end_judge(str)){
        return false;
    }
    transform(str.begin(), str.end(), str.begin(), ::toupper);

    if (is_number(str) && !cityList.empty()) {
        int num = stoi(str);
        if (num < 0 || num >= choose_City.size()) {
            cout << "please enter a correct input" << endl;
            return true;
        }
        *pt = choose_City[num];
        choose_City.clear();
        return true;
    }
```

```
    if (str.size() < 3) {
        cout << "Too short, the City name at least 3 char." << endl;
        return true;
    }

    choose_City.clear();
    for (auto city = cityList.begin(); city != cityList.end(); city++) {
        if (city->name.rfind(str, 0) == 0) {
            if (choose_City.size() == 1) {
                cout << "Those city is match, input integer to choose city" << endl;
                cout << "0. " << choose_City[0]->name << ", " << choose_City[0]->country << endl;
                cout << "1. " << city->name << ", " << city->country << endl;
            } else if (choose_City.size() > 1) {
                cout << choose_City.size() << ". " << city->name << ", " << city->country << endl;
            }
            choose_City.push_back(&*city);

        }
    }

    if (choose_City.size() >= 2) {
        return true;
    }else if (choose_City.size() == 0){
        cout<<"NO CITY EXIST!"<<endl;
        return true;
    }
    *pt = choose_City[0];
    return true;
}
double findDistance(const city &city1, const 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;
}
```

Part 3 - Result & Verification

Test case #1: **文件不存在**

我们将代码定义中的
#define FILE_NAME "world_cities.csv"
修改成其他不存在的文件名，如
#define FILE_NAME "demo.txt"

output:
we can't open this file: demo.txt

```
road@LAPTOP-UIB8HK80:~/ProjectOfWSL/assignment/assignment2$ g++ A2.cpp &&./a.out
we can't open this file: demo.txt
```

Test case #2: **城市长度过短**

如果城市长度过短,我们将要求重新输入城市名字
input:
ch
output:
Too short, the City name at least 3 char.

Input "bye" to end the programmer! You need at least input three char
please input the first city name:

input:
chengdu
be
output:
Too short, the City name at least 3 char.
please input the second city name:

```
road@LAPTOP-UIB8HK80:~/ProjectOfWSL/assignment/assignment2$ g++ A2.cpp &&./a.out

-----------------------------------
Input "bye" to end the programmer! You need at least input three char
please input the first city name: ch
Too short, the City name at least 3 char.

-----------------------------------
Input "bye" to end the programmer! You need at least input three char
please input the first city name:
```

```
road@LAPTOP-UIB8HK80:~/ProjectOfWSL/assignment/assignment2$ g++ A2.cpp &&./a.out

-----------------------------------
Input "bye" to end the programmer! You need at least input three char
please input the first city name: chengdu
please input the second city name: be
Too short, the City name at least 3 char.
please input the second city name:
```

Test case #3: **未能发现城市**

未能发现城市时，我们将会要求重新输入城市名字

```
input:
beijing
arr

output:
NO CITY EXIST!
please input the second city name:
```

```
road@LAPTOP-UIB8HK80:~/ProjectOfWSL/assignment/assignment2$ g++ A2.cpp &&./a.out

------------------------------------
Input "bye" to end the programmer! You need at least input three char
please input the first city name: beijing
please input the second city name: arr
NO CITY EXIST!
please input the second city name:
```

Test case #4: **除去城市名字前后的多余空格**

```
input:
     beijing
shenzhen

output:
The distance between BEIJING and CHENGDU is 6719.38 km.

------------------------------------
Input "bye" to end the programmer! You need at least input three char
please input the first city name:
```

```
road@LAPTOP-UIB8HK80:~/ProjectOfWSL/assignment/assignment2$ g++ A2.cpp &&./a.out

------------------------------------
Input "bye" to end the programmer! You need at least input three char
please input the first city name:         beijing
please input the second city name: chengdu
The distance between BEIJING and CHENGDU is 6719.38 km.

------------------------------------
Input "bye" to end the programmer! You need at least input three char
please input the first city name:
```

Test case #5: **模糊匹配并退出**

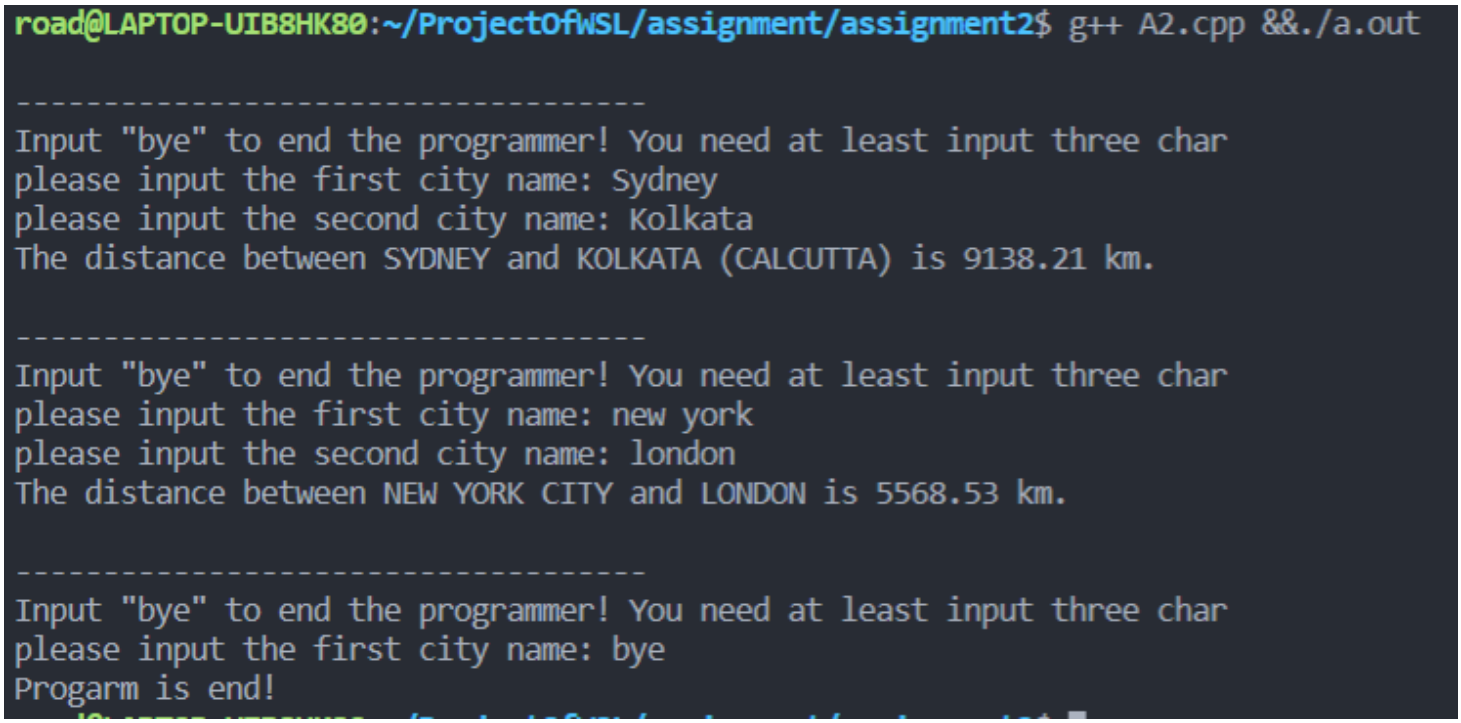
```
input:
bei
0
shanghai
bye
output:
Those city is match, input integer to choose city
0. BEIJING, China
1. BEIRUT, Lebanon
The distance between BEIJING and SHANGHAI is 1071.29 km.
Program is end!
```

```
road@LAPTOP-UIB8HK80:~/ProjectOfWSL/assignment/assignment2$ g++ A2.cpp &&./a.out

-----------------------------------
Input "bye" to end the programmer! You need at least input three char
please input the first city name: bei
Those city is match, input integer to choose city
0. BEIJING, China
1. BEIRUT, Lebanon

-----------------------------------
Input "bye" to end the programmer! You need at least input three char
please input the first city name: 0
please input the second city name: shanghai
The distance between BEIJING and SHANGHAI is 1071.29 km.

-----------------------------------
Input "bye" to end the programmer! You need at least input three char
please input the first city name: bye
Progarm is end!
```

Test case #6: **正常输入并退出程序**

这个正常的案例包括了名字中间的空格，大小写，多个case案例以及正常退出

```
input:
Sydney
Kolkata
New York
London

output:
he distance between SYDNEY and KOLKATA (CALCUTTA) is 9138.21 km.
The distance between NEW YORK CITY and LONDON is 5568.53 km.
Program is end!
```



Part 4 - Difficulties & Solutions

1. 难点1：文件写入并处理异常。处理：学习fstream用法进行文件的写入
2. 难点2：指针的使用与动态分配。处理：看c++primplus和课件进行学习与解决
3. 难点3： 模糊匹配。处理：循环处理比较string字符串放入choose_city中方便比较调用，并用其所在位置作为编号，方便预览与使用
4. 难点4： 将给定的"world_cities.csv"文件拷贝到WSL的服务器下。处理：上网查找资料，在wsl服务端上找到windows系统下的文件，使用cp这个指令拷贝到WSL服务器上