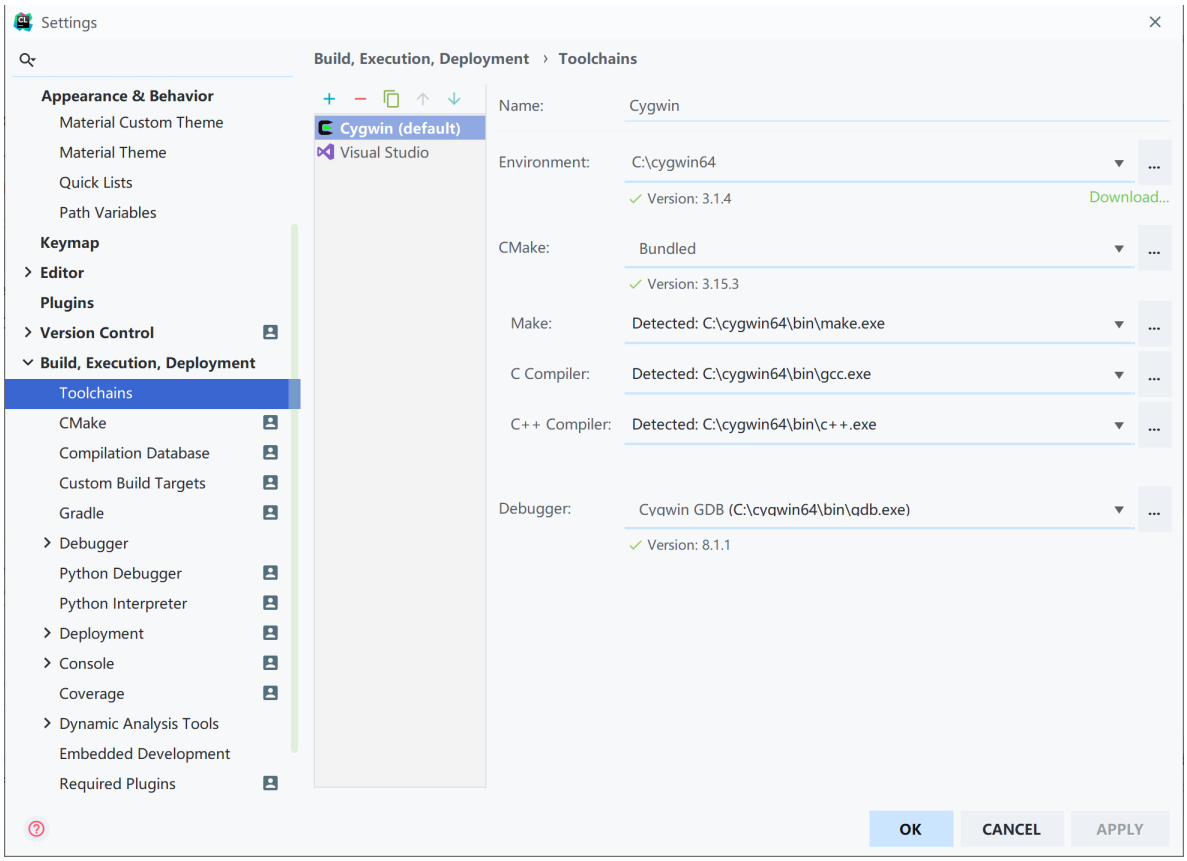
CS205 C/C++ Programming - Lab Assignment 1

Name: 蔡永宁(Cai Yongning)

SID: 11710802

environment:

coding on *CLion, Cygwin, window 10*



compile and run on *Window Subsystem Linux, Ubuntu, VS Code*

Part 1 - Analysis

Step1: program must store the information of city, so create a `struct city`

Step2: program should not crash when the input format is incorrect，so using `string` to store information, then check special characters as `void checkName(string)` and `void checkFloat(string)`

> reference: How can I check if a string has special characters in C++ effectively?

Step3: program next compute the flying distance between two city by `double calDistance(string, string, string, string)` using two cities' information

Step4: some tool function like `int main()` to get user's input, `double degreeToRad(double)` to change degree to rad, `double stringToFloat(string)` to convert string to float number, and `define PI 3.1415926535` for computing

> reference: std::string to float or double

Part 2 - Code

```
//
// created by Andyb1ance 2020/3/15
//
// assignment1: compute the flying distance between the two and display
//

#include <iostream>
#include <string>
#include <cmath>

using namespace std;
#define PI 3.1415926535
struct city {
    string city_name;
    string longitude;
    string latitude;
};

void checkName(string);

void checkFloat(string, string);

double stringToFloat(string);

double calDistance(string, string, string, string);

double degreeToRad(double);

int main() {
    city *first = new city;
    city *second = new city;
    //first city
    cout << "The first city: ";
    getline(cin, first->city_name);
    checkName(first->city_name);
    cout << "The latitude and longitude of first city: ";
    cin >> first->latitude;
    cin >> first->longitude;
    checkFloat(first->latitude, "la");
    checkFloat(first->longitude, "lo");
    //second city
    cin.get();
    cout << "The second city: ";
    getline(cin, second->city_name);
```

```
    checkName(second->city_name);
    cout << "The latitude and longitude of second city: ";
    cin >> second->latitude;
    cin >> second->longitude;
    checkFloat(second->latitude, "la");
    checkFloat(second->longitude, "lo");
    float res = calDistance(first->latitude, second->latitude, first->longitude,
second->longitude);
    cout << "The distance between " << first->city_name << " and " << second-
>city_name << " is " << res << " km";
    delete first;
    delete second;
    return 0;
}

void checkName(string name) {
    if
(name.find_first_not_of("abcdefghijklmnopqrstuvwxyzABCDEFGHIJKLMNOPQRSTUVWXYZ_
,") != string::npos) {
        cout << "find a mistake, the input format of city name is incorrect";
        exit(0);
    }
}

void checkFloat(string num, string type) {
    if (num.find_first_not_of("-1234567890.") != string::npos) {
        cout << "find a mistake, the input format of longitude or latitude is
incorrect";
        exit(0);
    }
    if (type == "la") {
        if (stringToFloat(num) < -90.0f || stringToFloat(num) > 90.0f) {
            cout << "find a mistake, the input out of latitude range";
            exit(0);
        }
    } else {
        if (stringToFloat(num) < -180.0f || stringToFloat(num) > 180.0f) {
            cout << "find a mistake, the input out of longitude range";
            exit(0);
        }
    }
}

double stringToFloat(string num) {
    return atof(num.c_str());
}

double degreeToRad(double degree) {
    return degree * PI / 180.0f;
}

double calDistance(string la1, string la2, string lo1, string lo2) {
    double phi1 = degreeToRad(90 - stringToFloat(la1));
    double phi2 = degreeToRad(90 - stringToFloat(la2));
    double theta1 = degreeToRad(stringToFloat(lo1));
    double theta2 = degreeToRad(stringToFloat(lo2));
    double c = sin(phi1) * sin(phi2) * cos(theta1 - theta2) + cos(phi1) *
cos(phi2);
```

```
    double d = 6371 * acos(c);
    return d;
}
```

Part 3 - Result & Verification

Test case #1

```
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 116.3917
The distance between Shenzhen and Beijing is <nearly 1942> km
```

```
andyb1ance@LAPTOP-M482O4RG:/mnt/d/code/cpp/SUSTech_CS205_cpp/assignment1$ g++ -o assignment1 main.cpp
andyb1ance@LAPTOP-M482O4RG:/mnt/d/code/cpp/SUSTech_CS205_cpp/assignment1$ ./assignment1
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 116.3917
The distance between Shenzhen and Beijing is 1942.84 km
```

Test case #2

```
Input:
New York, USA
40.7127 -74.0059
Rio de Janeiro, Brazil
-22.9083 -43.1964
Ouput:
The first city: New York, USA
The latitude and longitude of first city: 40.7127 -74.0059
The second city: Rio de Janeiro, Brazil
The latitude and longitude of second city: -20.9083 -43.1964
The distance between New York, USA and Rio de Janeiro, Brazil is <nearly 7555>
km
```

```
andyb1ance@LAPTOP-M482O4RG:/mnt/d/code/cpp/SUSTech_CS205_cpp/assignment1$ ./assignment1
The first city: New York, USA
The latitude and longitude of first city: 40.7127 -74.0059
The second city: Rio de Janeiro, Brazil
The latitude and longitude of second city: -20.9083 -43.1964
The distance between New York, USA and Rio de Janeiro, Brazil is 7555.98 km
```

Test case #3

```
Input:
New York; USA
Output:
<prompt and exit>
```

```
andyb1ance@LAPTOP-M482O4RG:/mnt/d/code/cpp/SUSTech_CS205_cpp/assignment1$ ./assignment1
The first city: New York; USA
find a mistake, the input format of city name is incorrect
```

Test case #4

```
Input:
New York, USA
-99 190
Ouput:
<prompt and exit>
```

```
andyb1ance@LAPTOP-M482O4RG:/mnt/d/code/cpp/SUSTech_CS205_cpp/assignment1$ ./assignment1
The first city: New York, USA
The latitude and longitude of first city: -99 190
find a mistake, the input out of latitude range
```

Part 4 - Difficulties & Solutions

Use a math library `<cmath>` to calculate trigonometric functions