CS205 C/ C++ Programming - Lab Assignment 3

Name: 陈宇恒 (Chen Yuheng)

SID: 11711111

Part 1 - Analysis

First, read the csv file line by line to initialize different city structures one by one, those cities are stored in an array. By using the technics of string dealing, we can splite out different parts of a line to store them in the different parts of city structure.

Then prompt user to input the city names, after trimming the space in both ends of input, if the name matches the only city in the list, then choosing step is finished; if no any match, then prompt user to input again; if exist more than one matches, list all the matched cities and prompt user to choose the right city he or she want. After some complex wrong-case dealing, we can get the right city in the city array we constructed before and the cities could be used to calculate the distance.

After two cities are selected correctly, using fomulas in assignment 1 can calculate the distance easily. Output the outcome and repeat the user-input part above.

At any input case, if "bye" is input, the programme will exit.

All the input will be capitalized as well as when comparing the name, so the input is case-insensitive.

Part 2 - Code

```
1 //compiled by gcc 6.3.0 (MinGW.org GCC-6.3.0-1)
   #include <iostream>
3 #include <fstream>
4 #include <string>
5 #include <string.h>
6 #include <sstream>
   #include <math.h>
9
   #define MAX_NAME_LENGTH 35
10
   #define ARRAY_SIZE 1000
11
12
   #define PI 3.1415926
13
   #define R 6371
14
    using namespace std;
15
16 | struct city{
17
        bool notNull=false;
18
        string name;
19
        string province;
        string country;
20
```

```
double latitude:
21
22
        double longitude;
23
    };
24
25
    void splite(string s, string strs[]);
    string truncate(string oldstr);
26
27
    void splite(string s, string strs[]);
28
    string up(string o);//capitalizing
29
    string& trim(string &s);//remove the space in both ends
    city chooseCity(city cities[], int n, string nd);//interaction with users to choose
30
    the right city
    double getDist(city city1,city city2);//computing
31
32
    city setCity(string strs[5]);//initializing the city objects
33
34
35
    template <class Type>
36
    Type stringToNum(const string& str){
37
        istringstream iss(str);
38
        Type num;
39
        iss >> num;
40
        return num;
41
    }
42
43
    string truncate(string oldstr){
        if(oldstr.length()<=MAX_NAME_LENGTH)</pre>
44
45
            return oldstr;
        cout << "WARNING: The string " << oldstr << " exceeds the max length of " <<</pre>
46
    MAX_NAME_LENGTH << ", it was truncated." << endl;</pre>
47
        return oldstr.substr(0,MAX_NAME_LENGTH);
48
    }
49
    void splite(string s, string strs[]){
50
51
        string delimiter = ",";
        int pos = 0;
52
53
        int i=0;
        while (i<4) {
54
55
            pos=s.find(delimiter);
56
            string token = s.substr(0, pos);
57
            strs[i]=token;
            s.erase(0, pos + 1);
58
59
            i++;
        }
60
61
        strs[4]=s;
62
    }
63
64
    string up(string o){//capitalizing
65
        char* arr=(char*) o.data();
66
        for(int i=0;i<0.length();i++){</pre>
67
            if(arr[i]>=97&&arr[i]<=122)
                arr[i]-=32;
68
69
70
        string n=arr;
71
        return n;
```

```
72
     }
 73
 74
     string& trim(string &s) {//remove the space in both ends
 75
         if (s.empty())
 76
             return s;
         while(s.find_first_not_of(" ")!=0||s.find_first_not_of("\t")!=0){
 77
 78
             s.erase(0,s.find_first_not_of(" "));
 79
             s.erase(0,s.find_first_not_of("\t"));
         }
 80
         while(s.find_last_not_of("
 81
     ")!=s.length()-1||s.find_last_not_of("\t")!=s.length()-1){
             s.erase(s.find_last_not_of(" ") + 1);
 82
 83
             s.erase(s.find_last_not_of("\t") + 1);
 84
         }
 85
         return s;
 86
     }
 87
 88
     double getDist(city city1,city city2){//computing
 89
         double phi1 = (PI/180) * (90 - city1.latitude);
 90
         double phi2 = (PI/180) * (90 - city2.latitude);
 91
         double theta1 = (PI/180) * city1.longitude;
 92
         double theta2 = (PI/180) * city2.longitude;
         double c = sin(phi1)*sin(phi2)*cos(theta1-theta2) + cos(phi1)*cos(phi2);
 93
 94
         double d = R*acos(c);
 95
         return d;
 96
     }
 97
 98
     city setCity(string strs[5]){//initializing the city objects
 99
         city ctemp;
100
         ctemp.notNull=true;
101
         ctemp.name=truncate(strs[0]);
102
         ctemp.province=truncate(strs[1]);
103
         ctemp.country=truncate(strs[2]);
104
         ctemp.latitude=stringToNum<double>(strs[3]);
105
         ctemp.longitude=stringToNum<double>(strs[4]);
106
         return ctemp;
     }
107
108
     city chooseCity(city cities[], int n, string nd){//interaction with users to choose
109
     the right city
110
         while(1){
111
             char c2[256];
112
             cout << "\nInput the name of the "<< nd <<" city: \n";</pre>
113
             cin.getline(c2,256);
             while(c2[0]=='\setminus 0')
114
115
                  cin.getline(c2,256);
116
             string c1=c2;
117
             c1=up(c1);
118
             c1=trim(c1);
119
             if(c1==up("bye"))
120
                  exit(0);
121
             if(c1.length()==0){
122
                  continue;
```

```
123
              }
124
              if(c1.length()<3){
125
                  cout << "\nIllegal input, please input the name of the "<<nd<<" city</pre>
     again: \n";
126
                  continue;
127
              }
128
              int count=0;
129
              city temp1;
              for(int i=0;i<n;i++){</pre>
130
131
                  if(up(cities[i].name).find(c1)!=string::npos){
132
                       count++;
133
                       temp1=cities[i];
134
                  }
135
              }
136
              if(count==0){
                  cout << "\nIllegal input, please input the name of the "<<nd<<" city</pre>
137
     again: \n";
138
                  continue;
139
              }
              if(count==1){
140
141
                   return temp1;
              }
142
143
              city temp_cities[count];
144
              int tempi=0;
              for(int i=0;i<n;i++){</pre>
145
146
                  if(up(cities[i].name).find(c1)!=string::npos){
147
                       temp_cities[tempi++]=cities[i];
148
                  }
149
              }
150
              cout << "\nPlease choose the city you want: \n";</pre>
              for(int i=0;i<count;i++){</pre>
151
152
                  cout << i+1 << ". " << temp_cities[i].name << ", "</pre>
     <<temp_cities[i].province << ", " <<temp_cities[i].country << ", "</pre>
      <<temp_cities[i].latitude << ", " <<temp_cities[i].longitude <<endl;</pre>
153
              }
154
              char* choose;
              cin.getline(choose,256);
155
156
              while(choose[0]=='\0')
157
                  cin.getline(choose,256);
              string choose_s=choose;
158
159
              choose_s=up(choose_s);
160
              choose_s=trim(choose_s);
161
              if(choose_s==up("bye"))
162
                  exit(0);
              int choose_n=stringToNum<int>(choose_s);
163
164
              while(choose_n<=0||choose_n>count){
165
                  cout<<"\nIllegal input, please input again: \n";</pre>
166
                  cin>>choose;
167
                  choose_n=stringToNum<int>(choose);
168
169
              return temp_cities[choose_n-1];
          }
170
171
     }
```

```
172
173
174
     int main(){
175
         city cities[ARRAY_SIZE];
176
         ifstream datas:
         datas.open("world_cities.csv", ios::in);
177
178
         if(datas.fail()){
             cout<<"WARNING: File is missing, please check it.\n";</pre>
179
             return -1;
180
181
         }
182
        int n;
         for(n=0;n<ARRAY_SIZE&&!datas.eof();n++){</pre>
183
184
            string temp;
            getline(datas,temp);
185
            char* char_array=(char*) temp.data();
186
187
            string strs[5];
188
            splite(temp,strs);
189
            cities[n]=setCity(strs);
190
191
         if(!datas.eof())
            cout << "WARNING: The number of lines exceeds the max number of " <<</pre>
192
     ARRAY_SIZE << ", the exceeds are not loaded." << endl;
193
         datas.close();
194
         while(1){
            city city1,city2;
195
            city1=chooseCity(cities,n,"first");
196
197
            city2=chooseCity(cities,n,"second");
198
            int d=(int) getDist(city1,city2);
199
     cout<<"The distance between "<<city1.name<<" and "<<city2.name<<" is "<<d<<"</pre>
200
     km."<<endl;</pre>
201
202
203
         return 0;
204
     }
```

Part 3 - Result & Verification

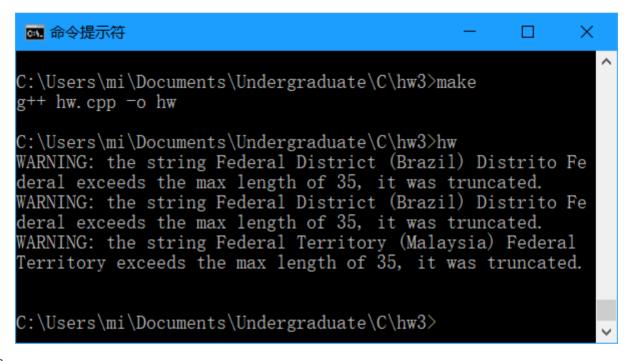
As shown, works well!

1_1:

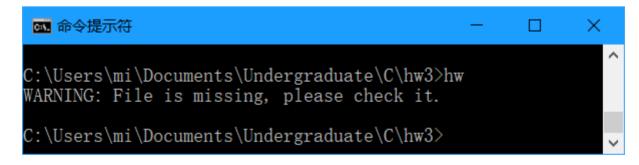
```
C:\Users\mi\Documents\Undergraduate\C\hw3>make
g++ hw.cpp -o hw

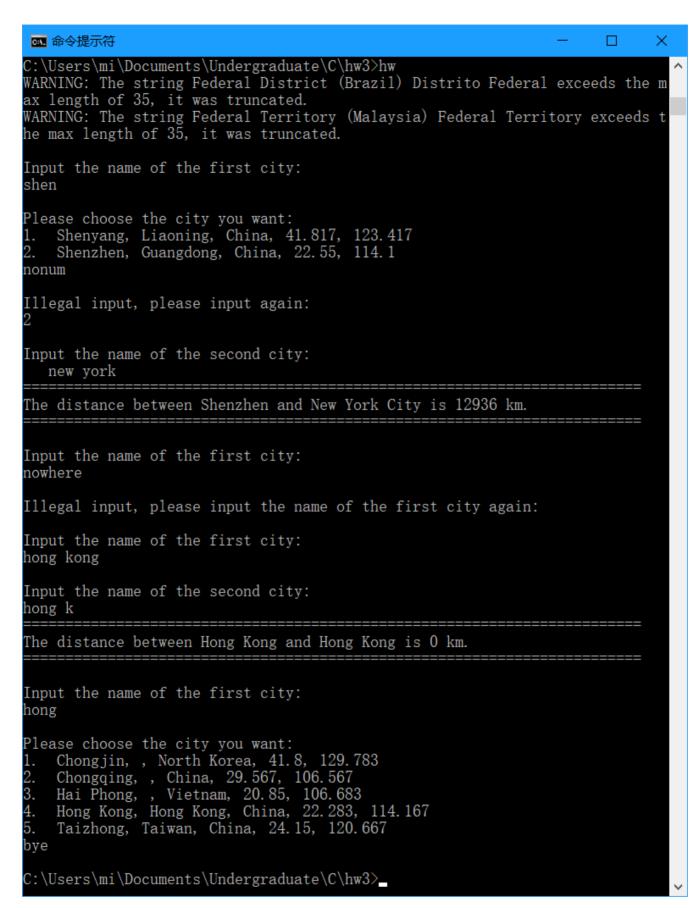
C:\Users\mi\Documents\Undergraduate\C\hw3>hw
WARNING: the string Federal District (Brazil) Distrito Federal exceeds the max length of 25, it was truncated.
WARNING: the string Federal District (Brazil) Distrito Federal exceeds the max length of 25, it was truncated.
WARNING: the string Australian Capital Territory exceeds the max length of 25, it was truncated.
WARNING: the string United States Virgin Islands exceeds the max length of 25, it was truncated.
WARNING: the string Islamabad Capital Territory exceeds the max length of 25, it was truncated.
WARNING: the string Saint Vincent and the Grenadines exceeds the max length of 25, it was truncated.
WARNING: the string Democratic Republic of the Congo exceeds the max length of 25, it was truncated.
WARNING: the string Las Palmas de Gran Canaria exceeds the max length of 25, it was truncated.
WARNING: the string Democratic Republic of the Congo exceeds the max length of 25, it was truncated.
WARNING: the string Democratic Republic of the Congo exceeds the max length of 25, it was truncated.
WARNING: the string Democratic Republic of the Congo exceeds the max length of 25, it was truncated.
WARNING: the string Federated States of Micronesia exceeds the max length of 25, it was truncated.
WARNING: the string Federated States of Micronesia exceeds the max length of 25, it was truncated.
WARNING: the string British Antarctic Territory exceeds the max length of 25, it was truncated.
WARNING: the number of lines exceeds the max number of 800, the exceeds are not loaded.
C:\Users\mi\Documents\Undergraduate\C\hw3>__
```

1 2:



1_3:





Part 4 - Difficulties & Solutions

- 1. I tried to printf to print a string, but the output is in a mess, I thought it is because the string is not correct, but later I found that it is because printf cannot be used to print string, cout is needed. Also, later I found that if cout and printf are used in one program, the output may be in a mess. At last, I use cout to output anywhere, the output is normal now.
- 2. To splite one line to seperate parts by comma is difficult, I tried to ways to do it, one is to skim two character array to seperate, but failed with confusing probleams, for certain lines with brackets the reading will stop, I don't know the reasons. Later I use substr and find function of string class to do the same task and succeeded.