# CS205 C/ C++ Programming Lab Assignment3


> Name: 黄玉安 (Huang Yu'an)
>
> SID: 11610303


## Part1 - Analysis

This assginment is a little complex and take me a little more time. The code for calcuating distance is the same as assignment 2 except the constant PI, I defined it by myself.

The first step is read information of cities from file, I use fstream. I use sstream to segment the semi for one line input and then put it in my City structure. Also, it need to check the file exist or if data are truncated.

Then I use a loop to let user input their choice, there will have hint to let them to input, and if they input wrong information (too short name, wrong index when result contain multi-city), I will let them repeate inputting until one city has been correctly found.

Some of the details were shown in Part2.

## Part2 -Code

Some of my definition

```
#define NAME_LEN 35
#define ARR_SIZE 1000
#define RES_INC_STEP 10
#define MIN(a, b) a<b?a:b
#define R 6371 // the redius of earth by km
#define PI 3.14159
```

```
// my structure:
struct City{
    char name[NAME_LEN];
    char country[NAME_LEN];
    double latitude;
    double longitude;
};
```

Parse the line in file:

```
int parse_line(string& line, City& city){
    int trancated_falg = 0;
    stringstream input_s(line);
    string temp;

    getline(input_s, temp, ',');
    memcpy(city.name, temp.c_str(), NAME_LEN);
    if(temp.length() > NAME_LEN){
        trancated_falg = 1;
```

```
    }

    getline(input_s, temp, ',');
    getline(input_s, temp, ',');
    memcpy(city.country, temp.c_str(), NAME_LEN);
    if(temp.length() > NAME_LEN){
        trancated_falg = 1;
    }

    getline(input_s, temp, ',');
    city.latitude = atof(temp.c_str());

    getline(input_s, temp, ',');
    city.longitude = atof(temp.c_str());

    return trancated_falg;
}
```

To found the first city:

```
   do{
            cout << "\nchoose the first city by input prefix of its name,
\ninput bye to exit:";
            getline(cin, comm);
            trim(comm);
            if(comm == "bye"){
                return 0;
            }
            city_index1 = get_city_choice(comm);
        }while (city_index1 <= 0);
        cout << "***choose first city:" << citys[city_index1].name << endl;
```

You can find more details in my code.

## Part 3 - Result & Verification

Case1:

> load data success!
> city number:988
>
> choose the first city by input prefix of its name,
> input bye to exit:New
> 1th city name:New Delhi
> 2th city name:New Orleans
> 3th city name:New York City
> 4th city name:Newcastle upon Tyne
> 5th city name:Newcastle
> choose the city by the inedx:4
> ***choose first city:Newcastle upon Tyne

```
choose the second city by input prefix of its name,
input bye to exit:New
1th city name:New Delhi
2th city name:New Orleans
3th city name:New York City
4th city name:Newcastle upon Tyne
5th city name:Newcastle
choose the city by the inedx:4
***choose second city:Newcastle upon Tyne
=====> the distance between Newcastle upon Tyne and Newcastle upon Tyne is:0

choose the first city by input prefix of its name,
input bye to exit:bye
```

Case 2: invalid input

```
choose the first city by input prefix of its name,
input bye to exit:New
1th city name:New Delhi
2th city name:New Orleans
3th city name:New York City
4th city name:Newcastle upon Tyne
5th city name:Newcastle
choose the city by the inedx:6
not have this index

choose the first city by input prefix of its name,
input bye to exit:Bye
not found this file

choose the first city by input prefix of its name,
input bye to exit:New
1th city name:New Delhi
2th city name:New Orleans
3th city name:New York City
4th city name:Newcastle upon Tyne
5th city name:Newcastle
choose the city by the inedx:3
***choose first city:New York City
```

The reslut is the same as what we want.

## Part 4 - Difficulties & Solutions

It is a little difficult to make it crash-free. I need to think invalid input case as many as possible and dea with them.