

C Language Report

jlu navigation system

2019/7/6

# Abstruct

This report will show you a C Language Program which called JLU Navigation System. In this exe, user can achieve four different functions, including inquiring information of a building, finding shortest path between two buildings, modifying building’s information like name, code and introduction and scanning all buildings’ names right now. In all functions, if user input wrong names, it will remind you that the name(s) you input is wrong and ask user to input again. What’s more, modifying information is the function that only administrators can do. And the other three functions are what user can use.

# 1.Problem statement

Like our name, our program is targeted at the fresh men of JLU in order to help them know the buildings abstractly. As the old saying goes” Beautiful Changchun city is located in JLU.”, JLU is actually huge and full of complex roads and various buildings. So I think our program may be of great use.

# 2.Group devision

|  |  |  |
| --- | --- | --- |
| Name | Student ID | College |
|  |  |  |
|  |  |  |
|  |  |  |

Yuchen Xu: Design the functions which include inquiring information of a building, modifying building’s information like name, code and introduction and scanning all buildings’ names right now. Link the functions in main function. Use file to store the information of different buildings. Set a background music for the program. Design the user interface with easyX and allow user to use mouse to choose the function they need.

Shichong Song: Use two-dimensional array to make map of JLU. Use different code to represent different buildings. Design the initial code for finding shortest path between two different buildings.

Ziyang Gao: Give the group the idea of the program. Supply different kinds of study resources.

# 3.Analysis

As the name of our program, our program is a system. As we all know, the interface and the interconnection with user is actually important in this situation. What’s more, as a navigation system, finding the shortest path between two different buildings is also of great significance. Our program contains many functions. The most important algorithm or function of the “JLU Navigation System” can be divided as the following 5 functions which are as follows.

## 3.1 login interface

When user firstly open this program, they will see this interface directly. In this interface, we should let users know what our program is and what they will find in this program. Then, as a system for a specific school, we can play the school music of this school to make users find this more fun.

## 3.2Choose interface

When users turn to this interface, we need to let them know what they can do in this program. In order to make them more convenient, we can use easyX to paint the rectangles with words inside to represent different functions. With the help of easyX, we can also allow users to choose the functions via using mouse. We can get the information of the location of mouse and then judge whether it is within the range of the rectangles.

## 3.3write & read buildinfo.txt

We can write this txt by using “fscanf”. We can put “|” between name and code, code and introduction. When we write next struct, we can put “\n”. For we write this txt in this specific format, we can read this file more easily to different structs.

For we want to remain the situation after modifying the information, so we should set up a txt to store the information. Then read the file character by character with the help of “fread”, and give what it gets to different structs.

## 3.4Write&read map.txt

We can write this txt in the excel. Use the method of two-dimensional array and different code for different buildings or for road, lake or grass. Then copy these data to a txt file and delete all tab in this txt file.

We can read this txt file by using “fscanf” to give the number it reads to an array. Then we can use it directly without “fscanf” again.

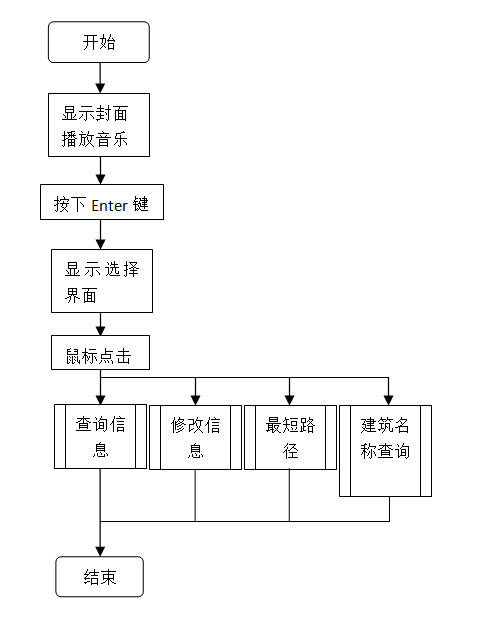
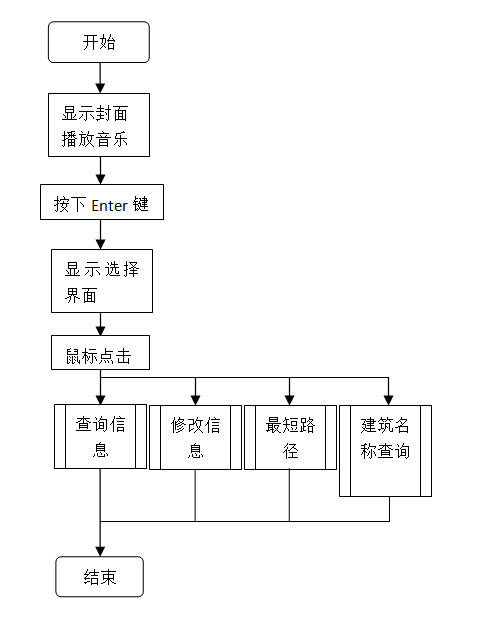
## 3.5 breadth-first search

To find the shortest path from the starting point to the ending point, we first find the path of the vertex adjacent to the starting point. Since the shortest path order of vertices is a breadth- first order, an auxiliary queue is needed. The specific steps are as follows:

1. From the starting point, which is the first to join the queue, set the distance to zero;
2. Take the position from the head of the queue, add the position from this position to the queue, and let the distance of these positions is the distance of the previous positions plus 1;
3. Loop 2 till the end, which means we have found the path;

In the process, the distance corresponding to each processing position is strictly increasing, so once the ending point is found, the distance is therefore the shortest.

# 4.Design



This is our initial design(5.1). This design was firstly come up by a member of our group. After group discussing, we all think this program is useful for freshman. Then, we started working on it. We searched the relative documents online. The algorithm is quite clear to be seen in some specific examples in the Internet.

**5.1**

We raised the question very often. Then we analyse and discuss over the whole topic, which made us determined to devote to the program. At last, we helped each other through tough times and successfully made it at last.

**5.2**

inquire information

Input the name of building you are inquiring

Search for the name you input and judge whether it exists

F

Print “wrong input, please retry”

T

Print the information stores in the document

The flow char above(5.2) is the design for the function called inquiring the information. Firstly, user needs to input the building they want to find. Then this application will get the building that users want to find and get the information from the “buildinfo.txt” character by character. Next, it will judge whether this name exists. If no, it will ask user to input again. If the building exists, it will output name and introduction of the building.

modify buildings

Input the administrator’s password

Judge the password is right or not

F

Print “wrong password”

T

Input the name(code,information) of building you wang to change

Print several information of the building you just choose

Input the sequence number of the information you want to change

Input new information

Store new information to the file again

**5.3**

This flow chart above(5.3) shows the function called “Modifying information”. For the reason that this function can only be used by administrator, so if user want to choose this function, they have to input the passport. If the passport is wrong, user cannot do this function. They had to close this function. If the passport is right, user can do this function. Firstly, it will ask user which building they want to modify. Then it will show the name, code and introduction of the building that user inputs and ask you which aspect the user want to modify. Next, the user needs to inputting the sequence of name, code or information and the new one. At last, it will tell you that you have successfully modified it.

**5.4**

Use BFS algorithm and input the shortest path

Paint the path in the picture

Shortest path

Input the name of two buildings you want to inquire

Search for the name you input and judge whether it exists

F

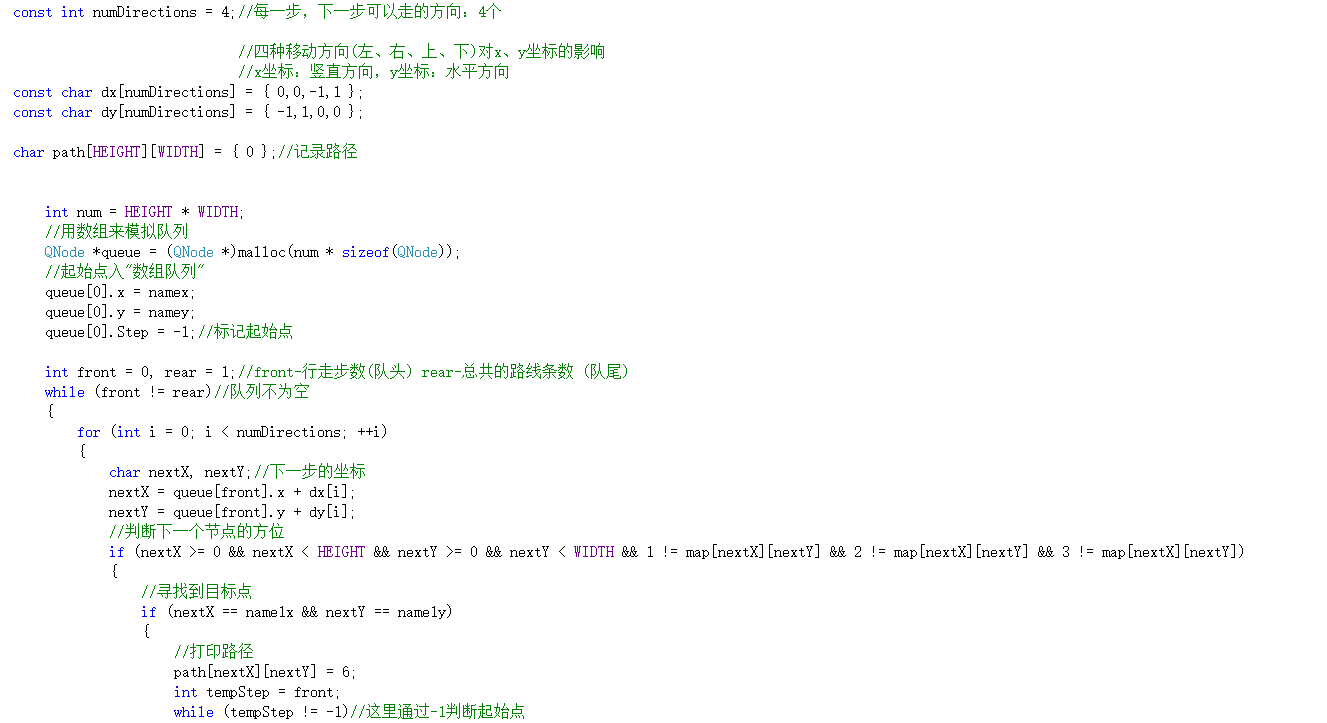
Print “wrong input, please retry”

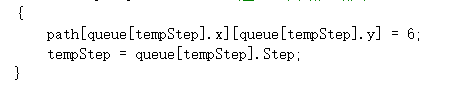
T

Read “map.txt” and translate the name into location

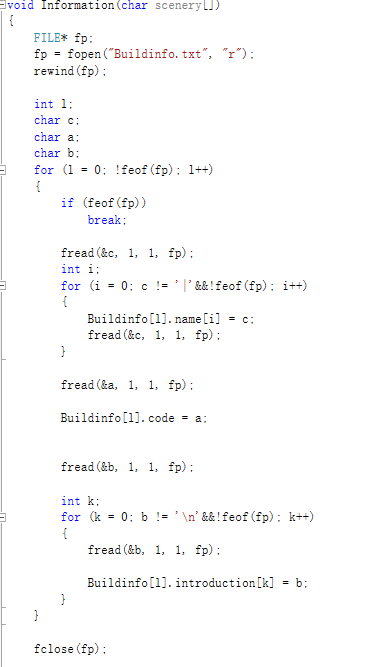
The flow chart above(5.4) shows the function called “shortest path”. User needed to input two different buildings at first. Then, it will judge whether they exist. If no, it will ask user to input again. If they exist, it will translate the name into their location in the map. Next, it will use BFS algorithm in order to output the shortest path and paint the road in different color.

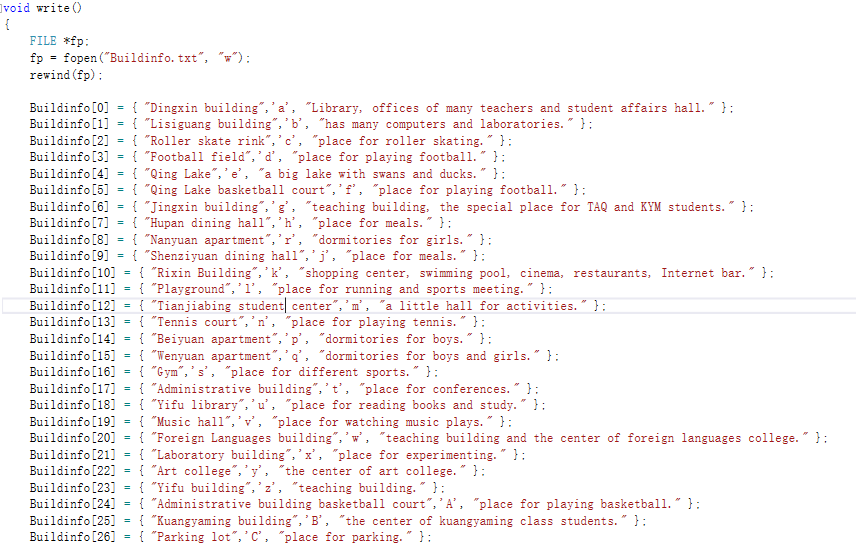
# 5.Implementation

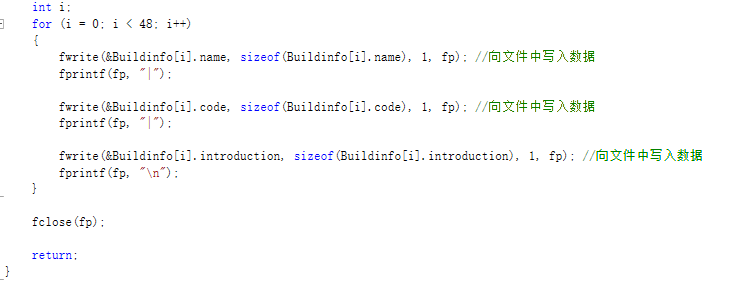
During the implementation part, we will put our design and analysis into real codes. Since the code is way too long, we would display some important part of the code to show our methodology. However, since the codes have been clearly displayed, further illustration would seem tedious and unnecessary. We would label the important parts of the codes, therefore benefit the readers.

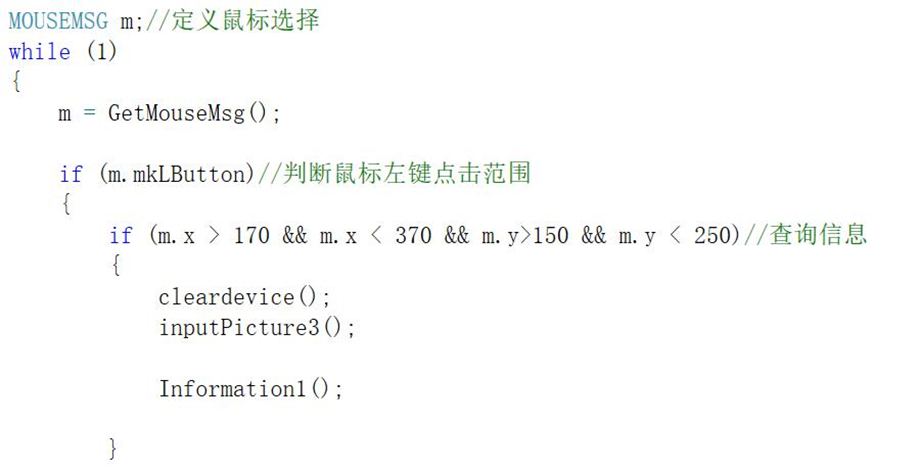




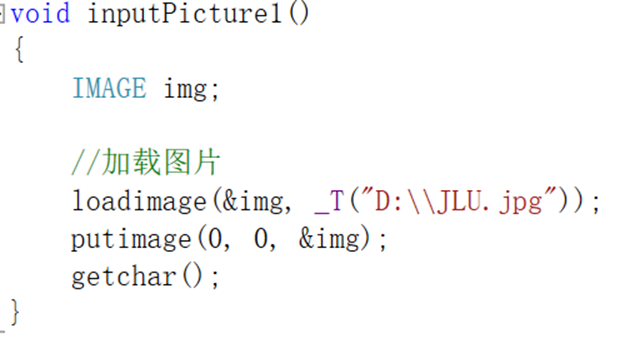
----Use of BFS algorithm

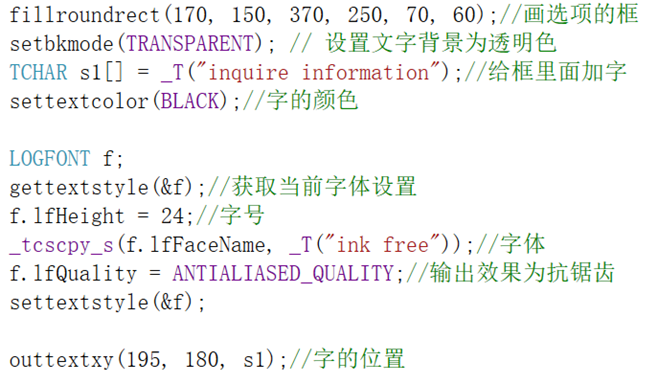




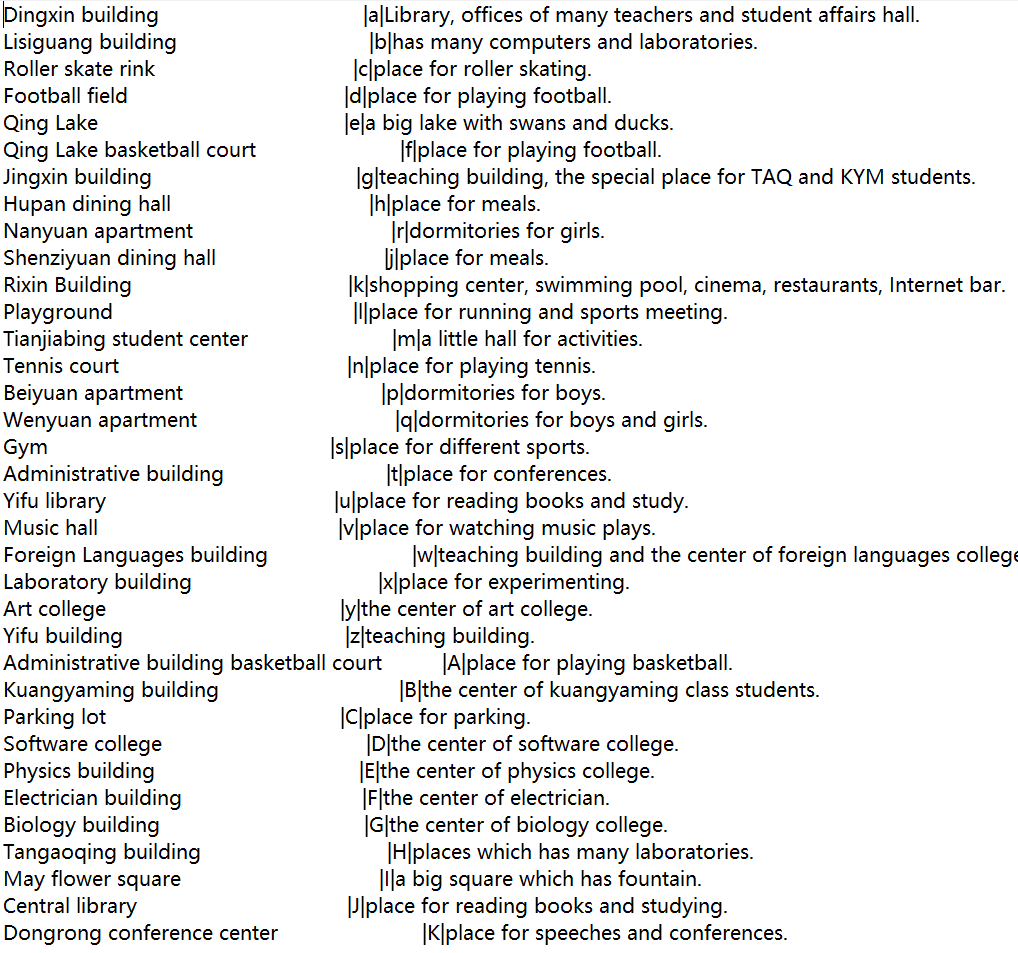
----Read “Buildinfo.txt” ----Write”Buildinfo.txt”

----Use mouse to choose the suitable function

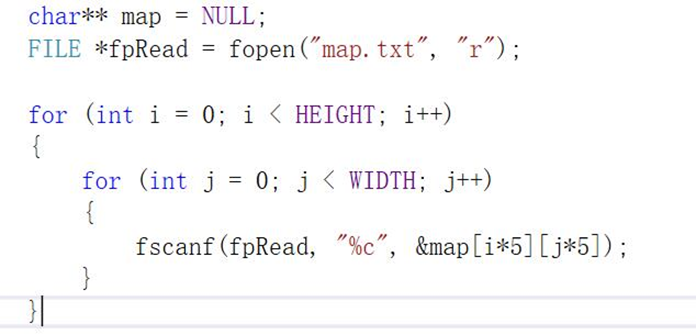


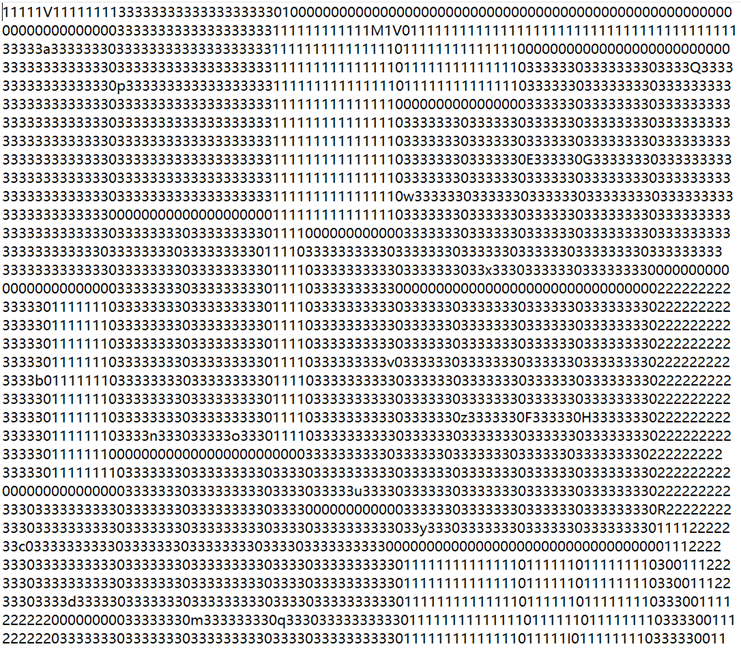
----User interface about uploading the picture 

----User interface about painting rectangle and put words in it



----Part about our “Buildinfo.txt”



 ----generate the map

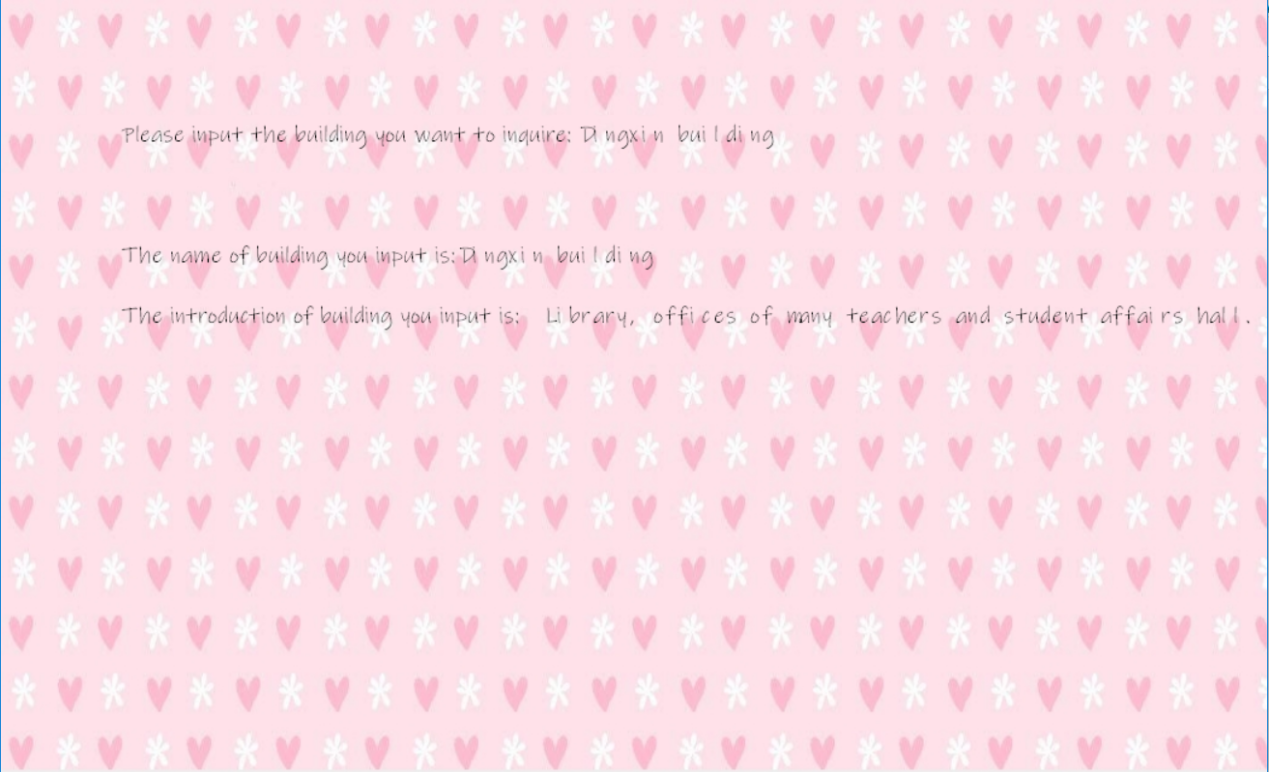
----Part of our map

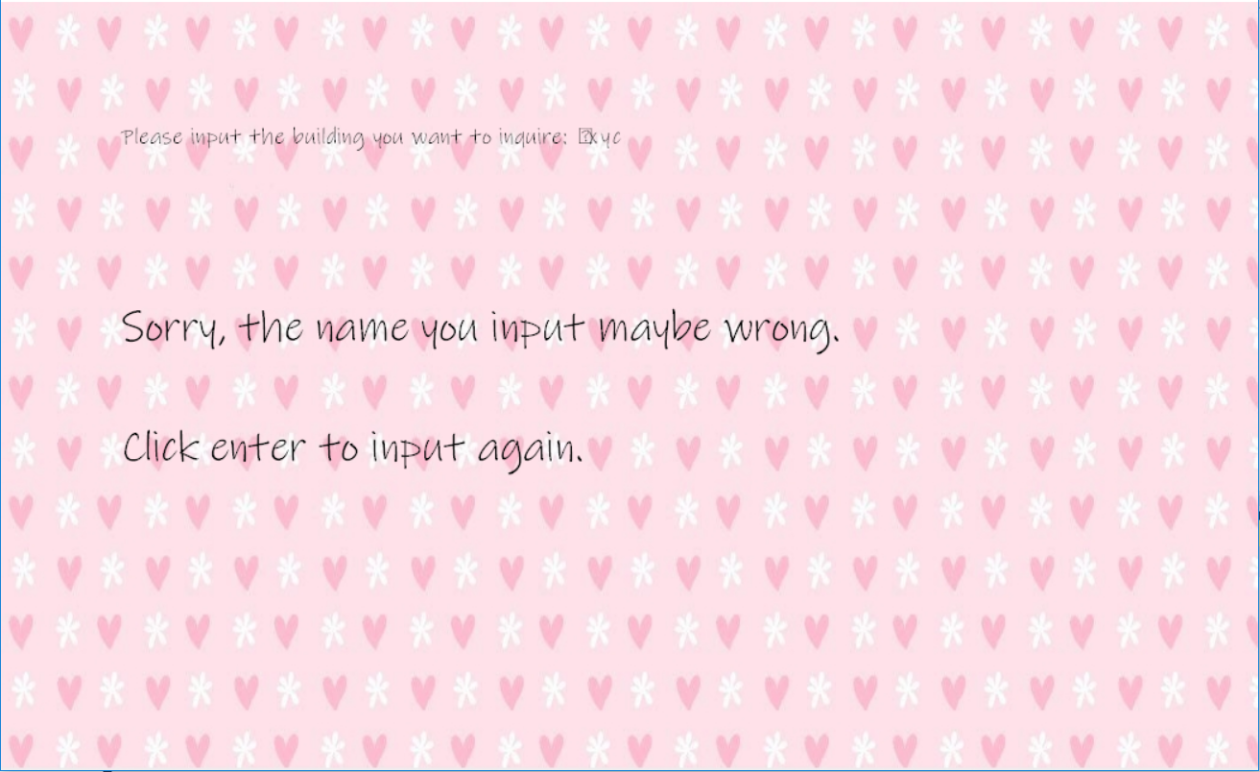
# 6.Testing and Debugging

(two interfaces in the beginning)

For the reason that our program has four different parts. So, our tests can be done separately.

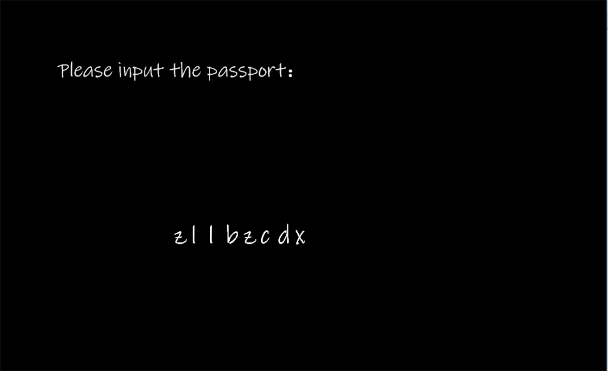
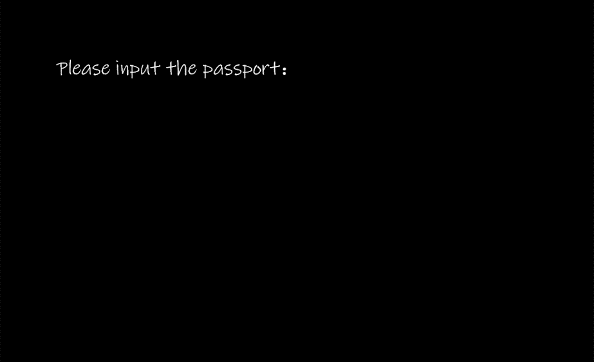
When I want to use file to store the modified information, I don’t understand the knowledge about file definitely. So, at that time, I learnt file from 0. With the help of senior students, I learnt it a lot and make the code on my own. (This is a testing screenshot about “Dingxin building”)



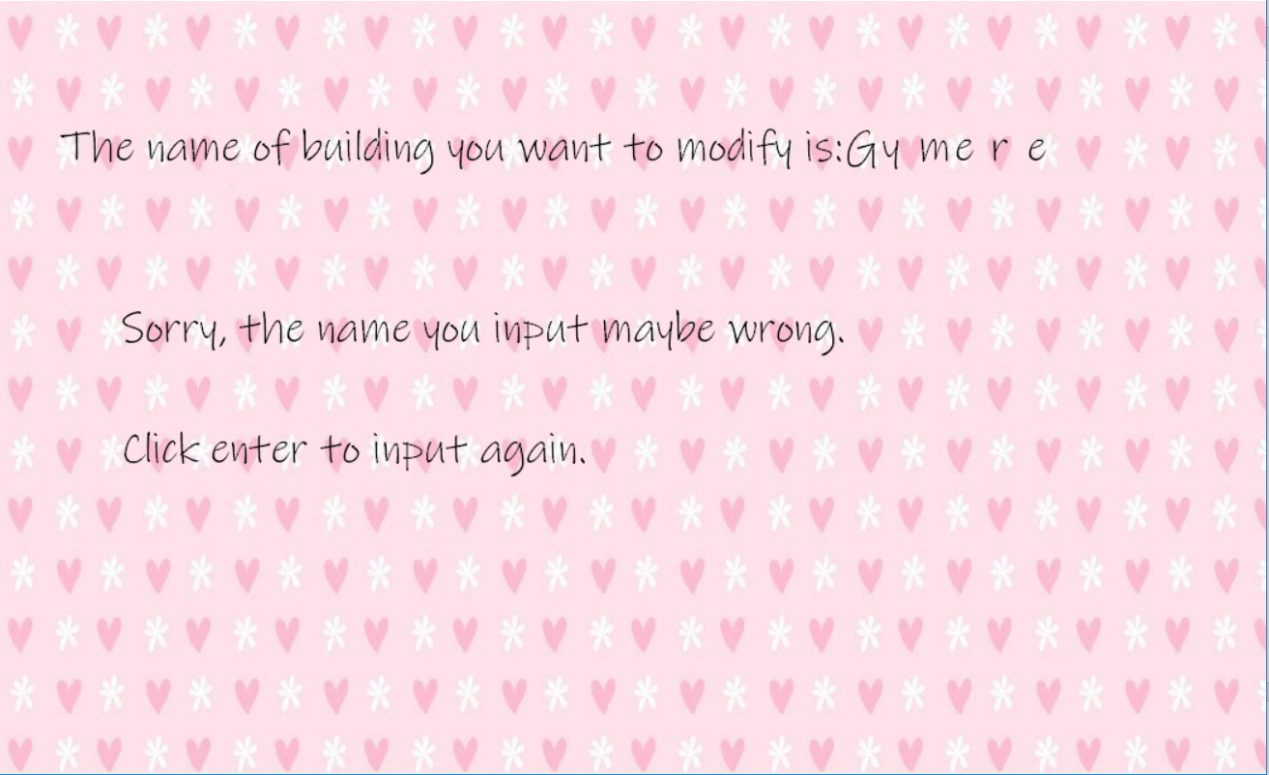
If the name you input is wrong, it will show this picture.

About modified the information, it needs to input passport which called ”zllbzcdx”. For the reason that I use easyX to make the user interface, I need to test where to show our words is best.

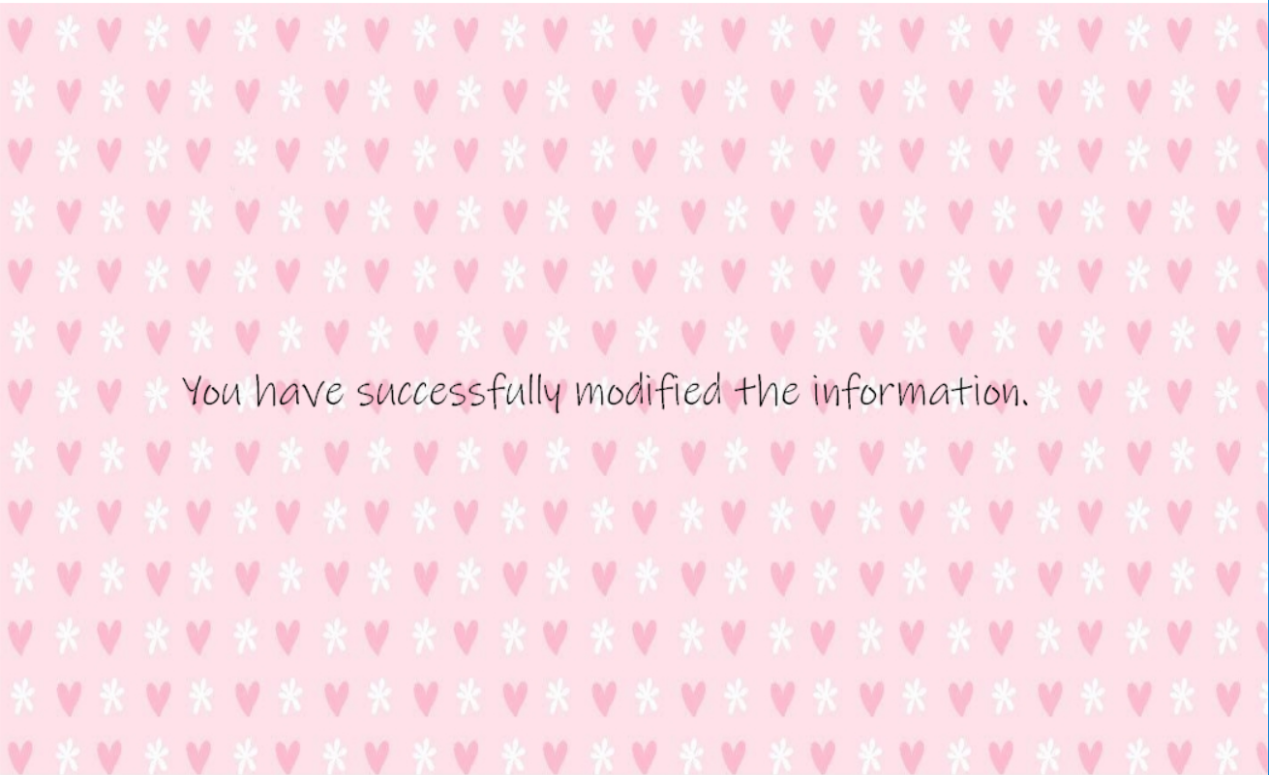
This is the screenshot about asking users to input passport.

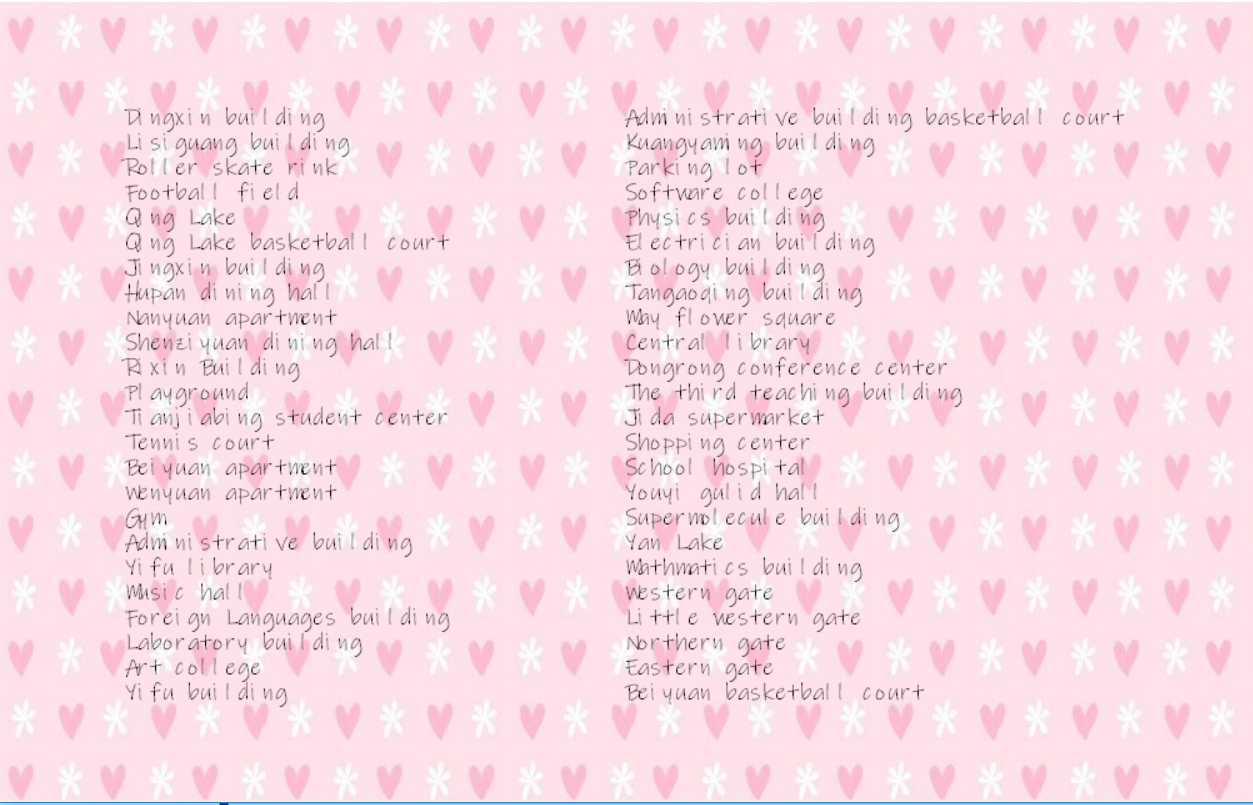


If the passport you input is right, you can turn to next page.

 If the name you input is wrong. And you can press Enter to input again.

 If the name is right. And then can change the thing you want to change.

At last, it will tell you that you changed successfully.

About scanning all the name. (It will be like this.)

# 7.Results and conclusion

During these day’s training, we can be confident to say that this program is like what we think it would be at the beginning. With the regular’s programming, compiling and debugging, we successfully grew up from a person who cannot solve a simple C Language questions to a person who can make a program. At the beginning time, I always feel restless for the difficult C Language program. But surprisingly, with the more and more study about C Language, I felt fun for making code which belongs me.

So, I think such program is of great fun and of great use. As the saying goes:” Experience is the best teacher”, I was so glad to see my transforming. With the cooperation with unknown person, I get friendship. And I know that everyone plays an important role in our system, so we can only make progress when everyone is doing their best for the group.

At first, the complement of our program cannot be achieved without the help of my group members. What’s more, in the process of making this program, I get many helps from Li nan sister, Meng Zhenyuan senior brother and Xing Hao senior brother. They always help me whenever I need help. I was so moved for their kindness. Most importantly, with the teaching form Xu Hao Professor and AT Professor, I know a lot in the field of C Language. I firmly think that these knowledge can be a fortune for my further study in Math.