

苏州大学

数据结构大作业实验报告

题 目 校园导航系统（独墅湖校园）

专 业 统计学

班 级 ****

学 号 *****

姓 名 杜冰

完成日期 2020 年 6 月 20 日

一 问题描述

设计一个独墅湖校区校园的导游系统，为来访的客人提供各种信息查询服务。查询服务有：提供任意景点/建筑物的信息，提供任意两点之间的最短路径，提供任意景点之间的所有路径。

基本功能

设计学校的校园平面图（可以通过调用学校的平面图完成景点的选取），要求景点/建筑不少于 12 个。以图中顶点表示校园内各景点/建筑，存放景点/建筑名称、代号、简介等信息；以边表示路径，存放路径长度等相关信息。

可以显示校园的平面图或者是所有景点/建筑

可以查看景点/建筑的相关信息介绍和周边信息（比如：与之最近的景点或者周边的道路信息）

可以查询任意两个景点/建筑之间的最短的简单路径。

可以查询任意两个景点/建筑之间的所有路径。

可以查询任意景点/建筑出发游览游览完其他景点/建筑的路线信息。

设计一个实用的查询界面和功能菜单（便于操作）

二 问题分析

根据基本功能要求可知，该系统面向游客共要完成 6 个功能，分别是显示所有建筑的名称及其编号、查询建筑的相关信息、查询任意两建筑之间的最短的简单路径、查询任意两个建筑之间的所有路径、查询从某个建筑出发游览完其它所有建筑的路线信息。除了面向游客的基本功能外，在系统内保存建筑的相关信息及路径信息。

因为建筑之间可以来回走动，所以采用无向图来保存相关信息，本系统采用无向图的邻接矩阵及邻接表保存建筑所有信息及路径。

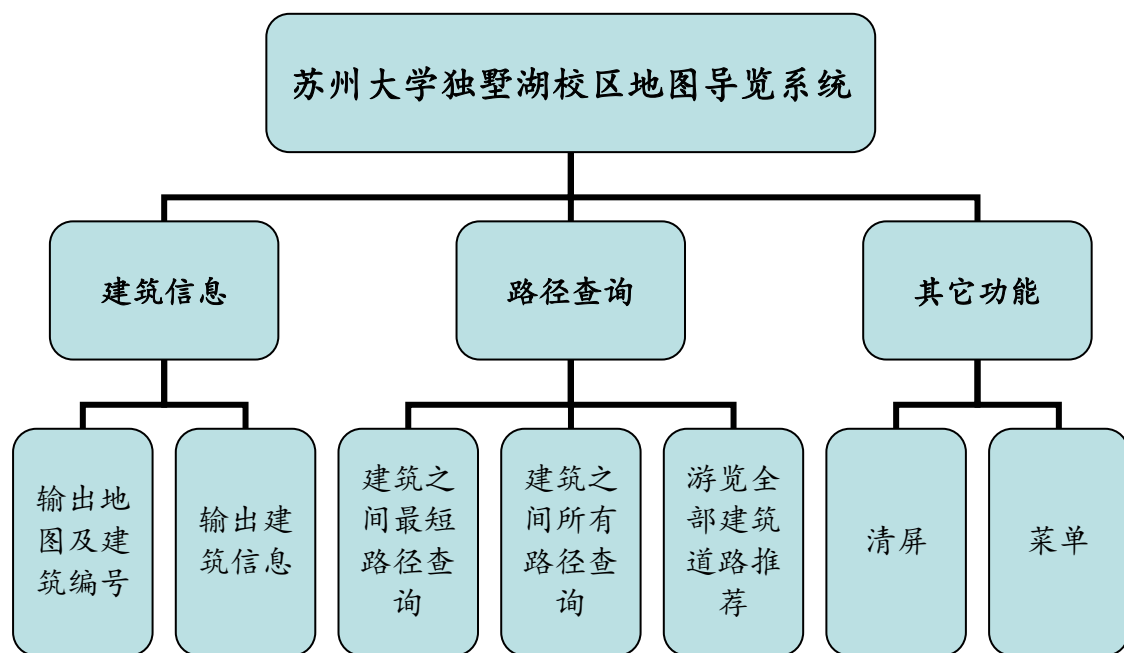
基本功能的具体实现分析如下：

1. 对于保存建筑的相关信息及路径信息，定义 `Init()` 函数，用来保存建筑信息；
2. 显示所有的建筑名称及编号可以直接进行输出，故直接将输出信息保存在了 `Output()` 函数中，同时定义了 `map()` 函数，用于输出地图；
3. 查询建筑的相关信息游客输入建筑的编号，根据编号可以知道对应保存相关信息的地址，将其输出即可，此功能用函数 `Introduce()` 实现；

4. 查询任意两建筑之间的最短路径可以转化成图的最短路径问题，故本题目使用 Floyd 算法求最短路径，然后将最短的路径使用 `Display(int i, int j)` 进行输出，将两个函数放到一起，就是我们的 `ShortDistance()` 函数；
5. 查询任意两个建筑之间的所有路径，可以转化成无向图中两个顶点之间的所有简单路径，参考课本第八章实验 10，编写了 `PathAll(AGraph* G, int u, int v, int path[], int d)` 函数，用于求顶点 `u` 到顶点 `v` 的所有的简单路径，然后使用函数 `AllDistance()` 完善功能，使系统具有更好的交互性，由于本系统中建筑以及路径都比较多，两点之间的所有路径都输出到屏幕上会令人眼花缭乱，故该算法中将所有路径保存到了文件中，可以打开相应文件查看两点之间的所有路径；
6. 查询任意景建筑出发游览游览完其他建筑的路线信息，该功能类似于旅行商问题，故该功能借鉴了动态规划法解决旅行商问题，编写了 `Tsp()` 函数实现功能，然后编写了 `GetPath()` 和 `DispPath()` 函数分别用于保存路径和输出路径，最后将这些函数整合放到了 `AdvicePath()` 函数中，本函数即为实现该功能的总函数，使用贪心算法同样能够解决该问题，但由于贪心算法容易陷入局部最优解，故未采用；题目还要求设计一个实用的查询页面和功能菜单，本题目使用 `System(int n, int &fl)` 函数建立了菜单系统，具体功能见下文中界面交互及功能分析。
7. 除了题目要求的功能外，本题目还为系统设计了密码，密码输入错误时无法进入系统，该功能用以对系统的使用权进行把控；同时该系统同时还提供了清屏功能，如果使用时间较长时感觉页面混乱，可以使用清屏功能，回到最初进入系统时的页面。

三 设计

● 主要功能模块划分



● 界面交互及功能分析

使用时输入密码（2020612）即可进入系统。

如果你想输出地图及所有建筑的编号，你可以输入数字 1，获取获取地图及建筑的编号；

如果你想知道建筑的详细信息，你可以输入数字 2，获取建筑的详细信息；

如果你想了解两个建筑之间的最短路线，你可以输入数字 3，获取两个建筑之间的最短路线；

如果你想了解两个建筑之间的全部路线，你可以输入数字 4，获取两个建筑之间的全部路线；

如果你想知道从某一建筑出发，游览全部路线的道路推荐，你可以输入数字 5，获取推荐路线；

如果你忘记了每个操作对应的数字，请不要着急，输入数字 6，即可获取系统菜单；

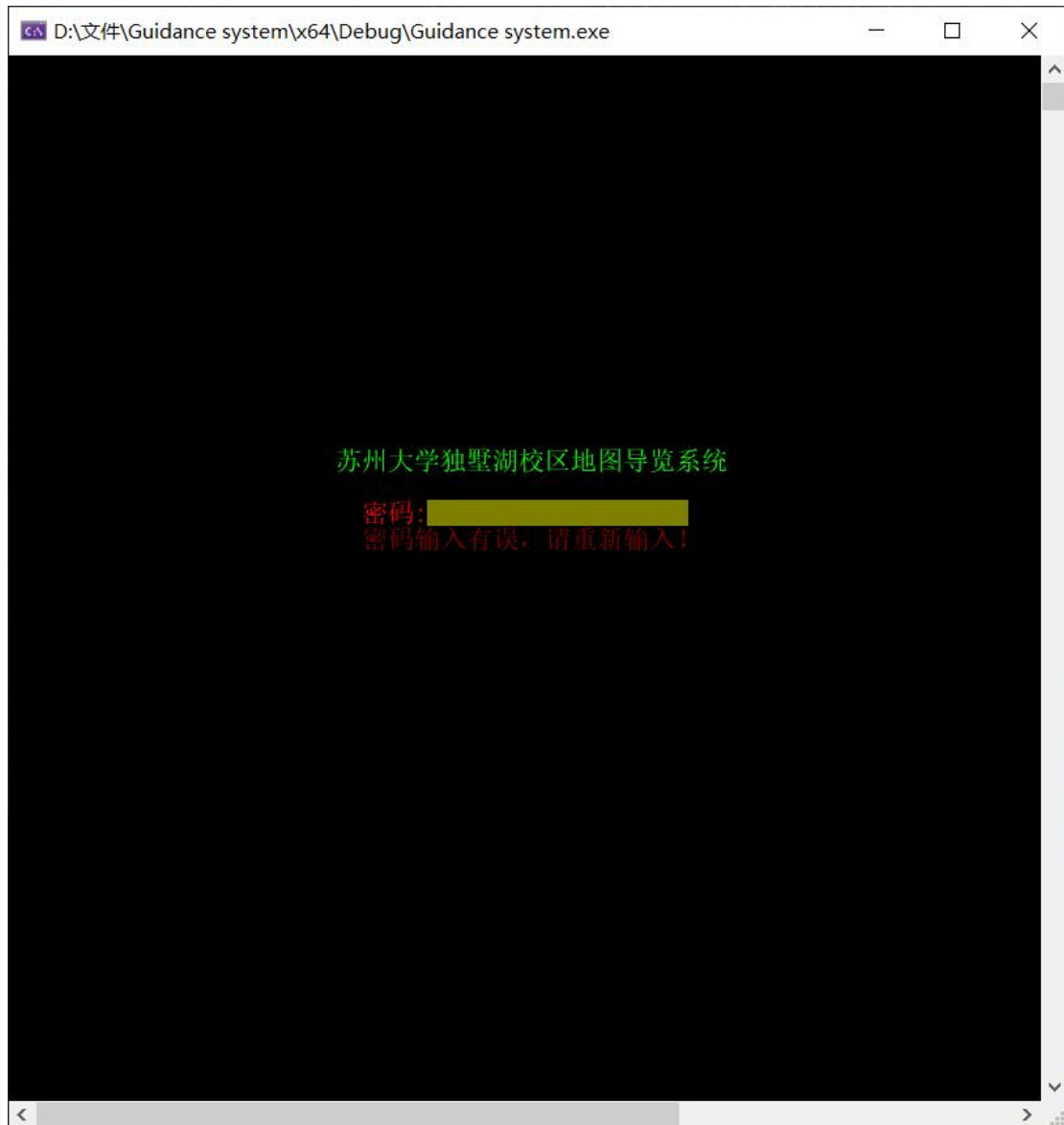
如果你感觉屏幕太满，你可以选择清屏选项，输入数字 7，即可进行清屏；

使用完想要退出程序，输入数字 0，即可退出程序。

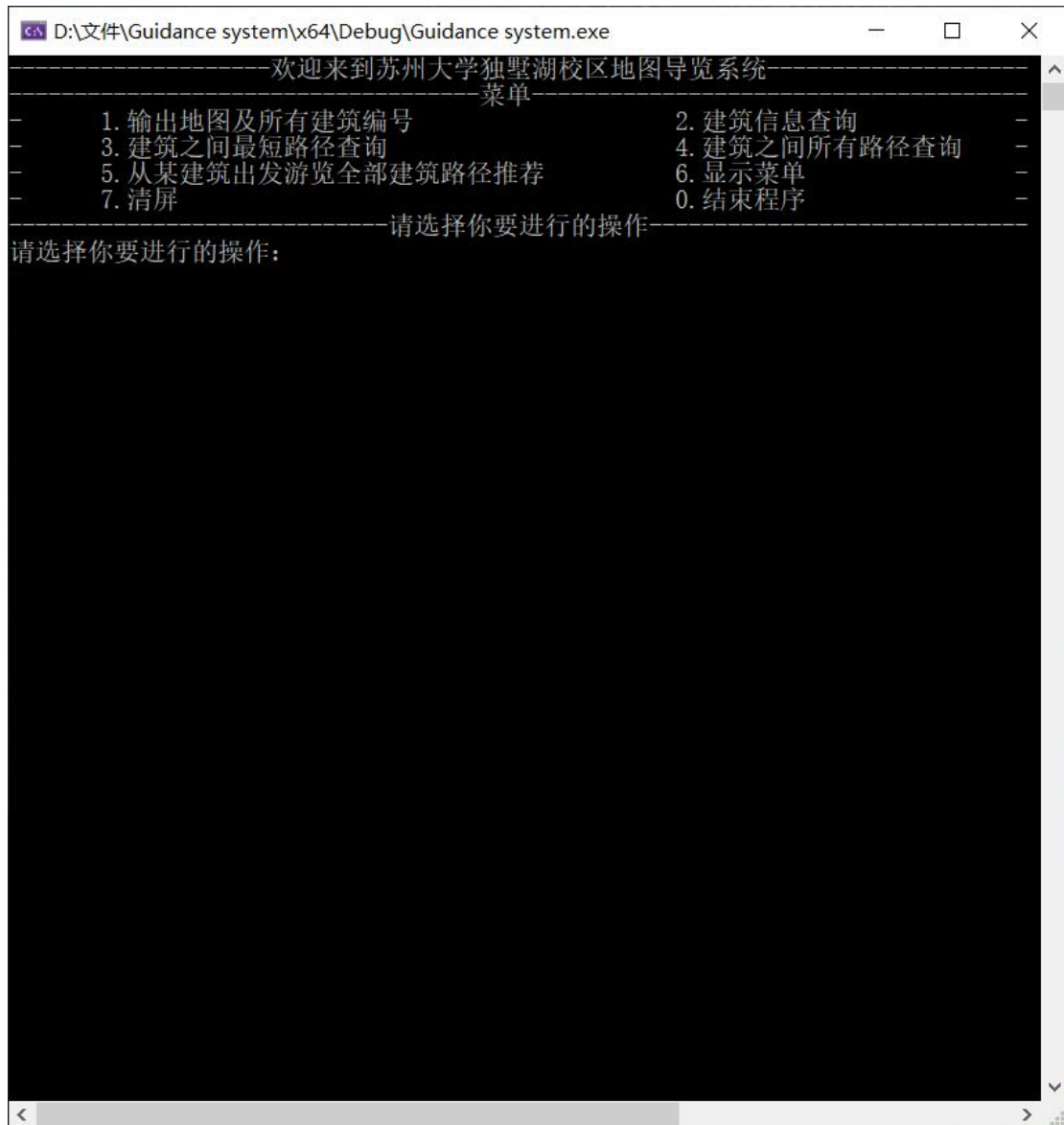
四 测试

● 不同的测试数据及输出结果

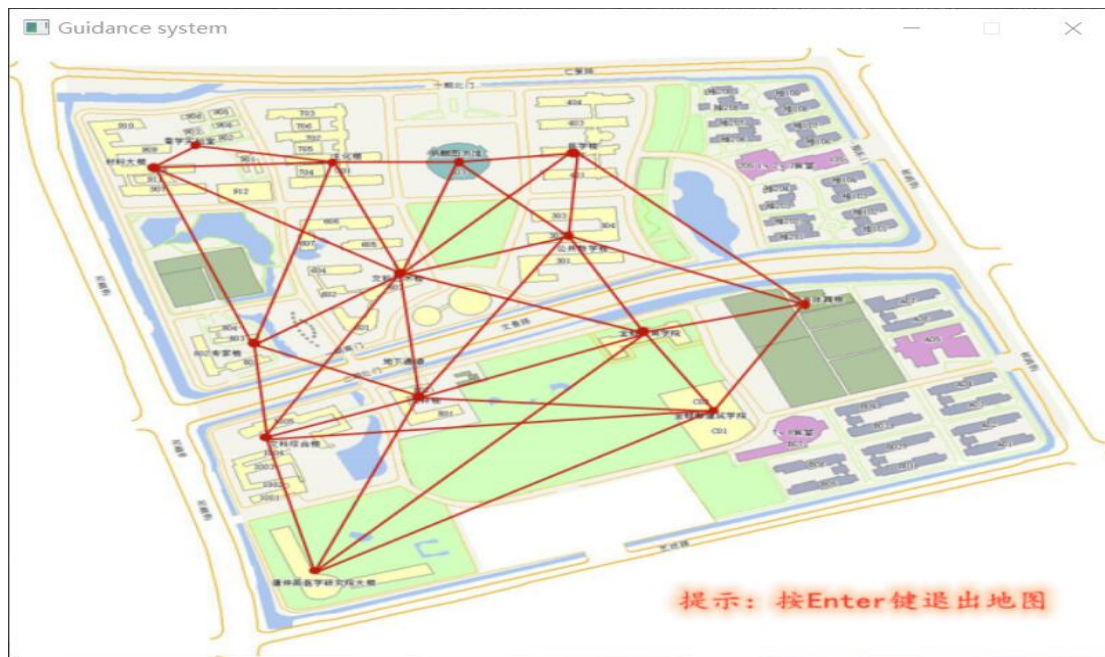
1.输入密码错误时界面



2.输入密码正确时进入的界面



3.进行操作 1 时弹出的地图及输出结果



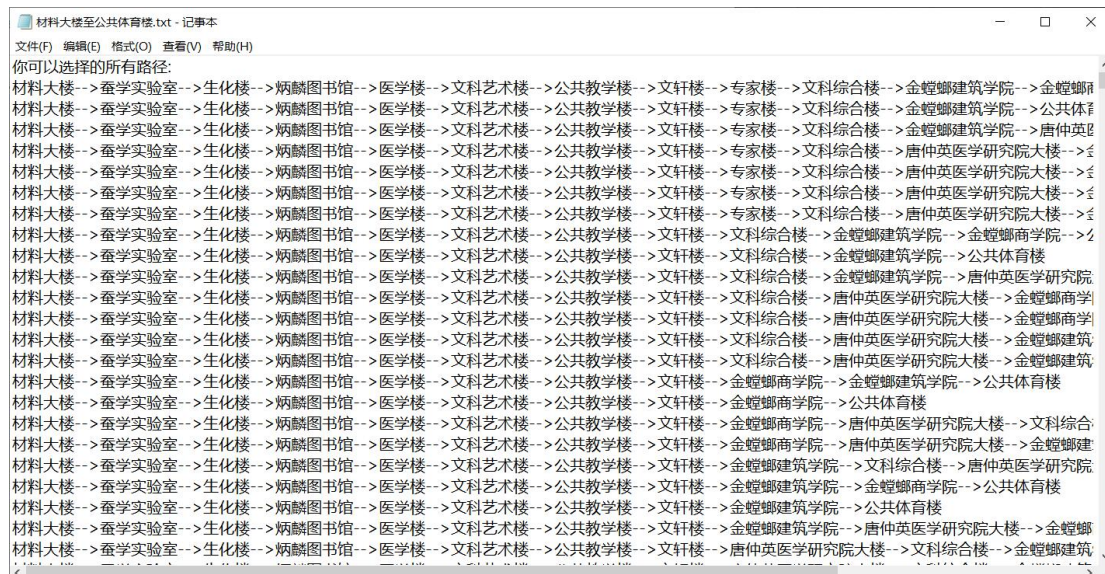
4.分别进行进行操作 2 3 时的运行结果



5.进行操作 4 时，等待运行结果时界面



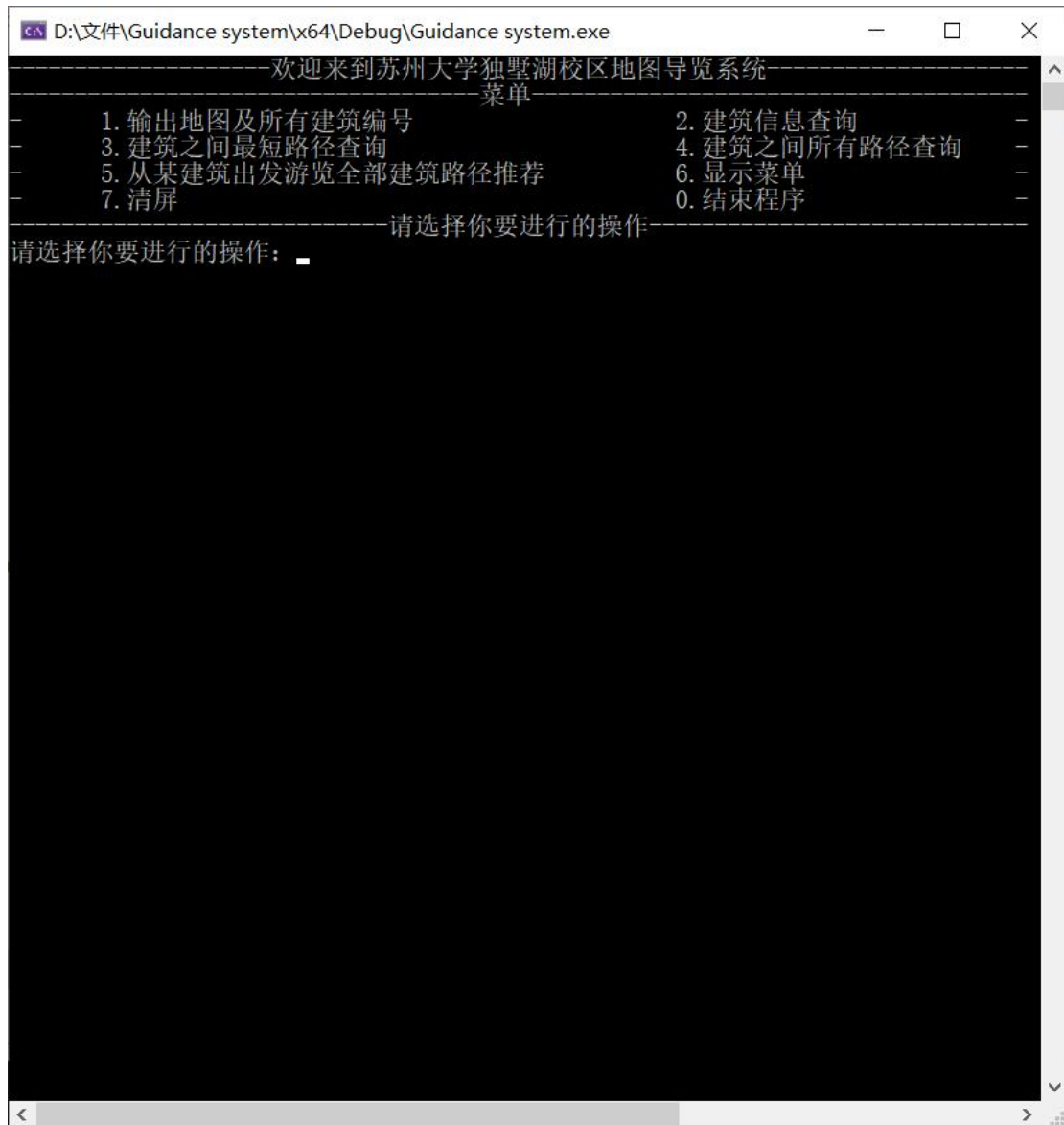
6. 进行操作 4，运行结果及获得的所有路线



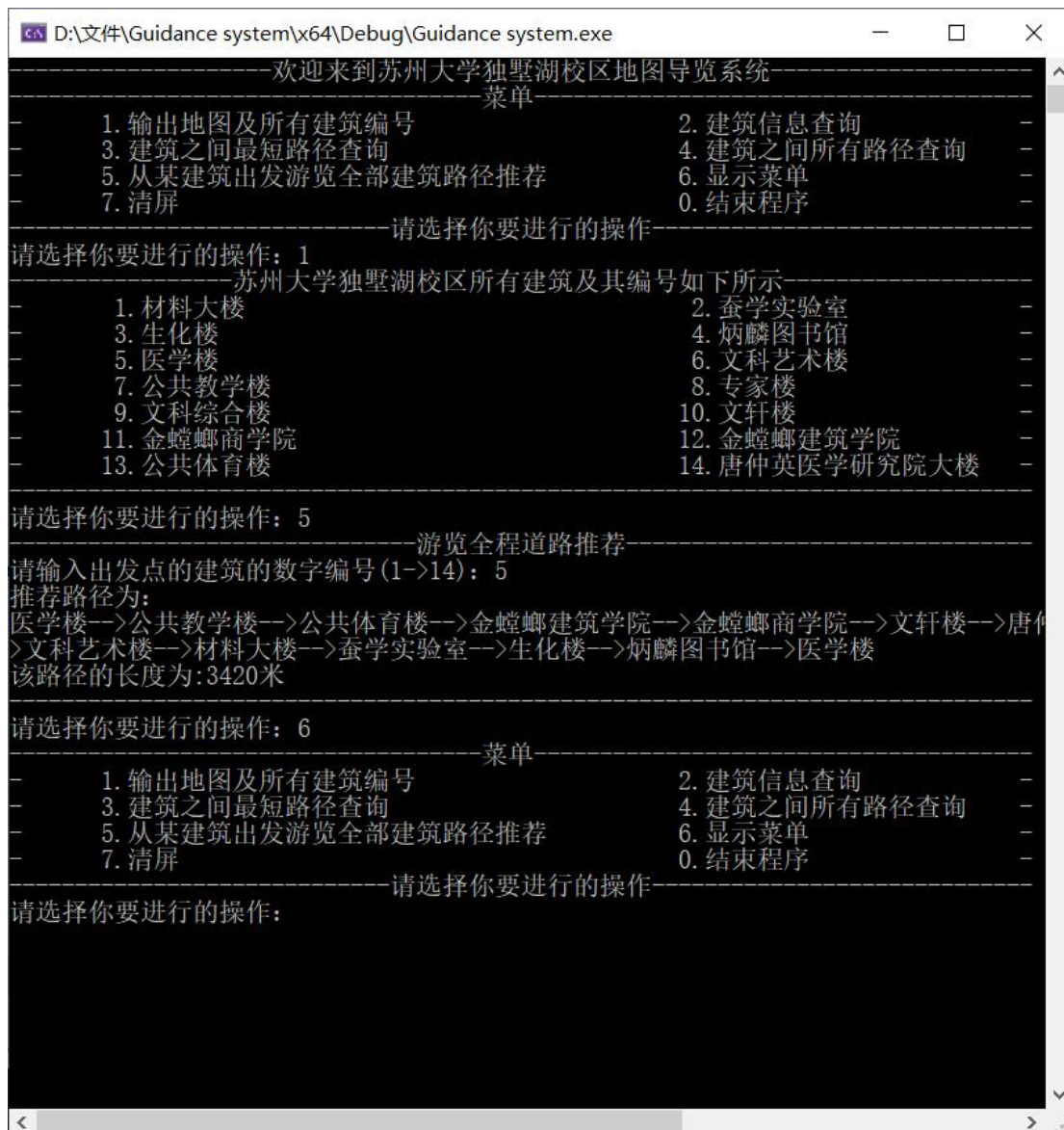
7.清屏前运行界面



8.清屏后运行界面



9.分别进行操作 5 6 时的运行结果



10.程序结束界面



五 总结

在设计系统时首先要列入一个框架，将每一个功能想一遍，应该如何完成这个功能，如何借助学到的知识去解决问题；在做系统时应根据列出的框架分别完成各个部分的算法，并测定是否能够完成自己预想得功能，在测试并完善好后再将其加入到系统中，否则会不知道哪里出现了问题，以及如何更改；总之，做大作业的过程是一个提高自己的过程，可以使自己对相关知识了解得更加透彻，同时锻炼自己，让自己了解到还有许多东西需要学习。

六 主要算法

● 设计部分

本系统采用无向图来保存相关信息,同时使用无向图的邻接矩阵及邻接表保存建筑所有信息及路径。

本系统定义了 Init() 函数,用来保存/初始化建筑信息;本系统使用 Output() 函数输出建筑及其编号,同时定义了 map() 函数, map() 函数中使用了 graphics.h 库中的函数,用于输出地图;定义了函数 Introduce() 实现对建筑进行介绍;本系统使用了 Floyd 算法求最短路径,然后将最短的路径使用 Display(int i, int j) 进行输出,然后将两个函数放到一起,就是我们的 ShortDistance() 函数,即为求两建筑之间最短路径的函数;为实现查询任意两个建筑之间的所有路径的功能,本系统编写了 PathAll(AGraph* G, int u, int v, int path[], int d) 函数,用于求顶点 u 到顶点 v 的所有的简单路径,然后使用函数 AllDistance() 完善功能,使系统具有更好的交互性,由于本系统中建筑以及路径都比较多,所以该算法中将所有路径保存到了文件中,可以打开相应文件查看两点之间的所有路径;查询从任意建筑出发游览游览完其他建筑的路线信息,该功能类似于旅行商问题,故该功能借鉴了动态规划法解决旅行商问题,编写了 Tsp() 函数实现功能,同时编写了 GetPath() 和 DispPath() 函数分别用于保存路径和输出路径,最后将这些函数整合放到了 AdvicePath() 函数中,得到查询从任意建筑出发游览完其它所有建筑路线的推荐。

● 算法部分

```
void Start(); //开始界面
void System(int n, int &fl); //菜单系统,n为输入的系统密码,flag作为密码是否输入正确的标志
void Menu(); //输出菜单
void Init(); //保存建筑信息
void map(); //输出地图
void Output(); //输出建筑名称及编号
void Introduce(); //建筑介绍
void Floyd(); //Floyd算法
void Display(int i, int j); //打印两个建筑的路径及最短路径,i,j分别为两建筑的编号
void ShortDistance(); //查找的两建筑的最短路径
void CreateAdj(AGraph*& G, int A[ViewNum][ViewNum], int n, int e); //创建图的邻接表
void MatToList(MGraph g, AGraph*& G); //将邻接矩阵g转换成邻接表G
void PathAll(AGraph* G, int u, int v, int path[], int d); //求顶点u到顶点v的所有的简单路径,u,v分别为两建筑的编号,path保存路径,d为path中的路径长度,初始为-1
void AllDistance(); //查找两建筑之间的所有路径
```

```

void TSP(); //求出动态规划dp数组
bool IsVisited(bool visited[]); //判断结点是否都以访问, 不包括0号结点
void GetPath(); //获取最优路径并保存在pathadvice中
void DispPath(); //输出路径
void AdvicePath(); //游览全程道路推荐
void ModSet(int w, int h); //设置窗口大小, w为宽度大小, h为高度大小
void setColor(unsigned short ForeColor, unsigned short BackGroundColor); //设置字体颜色, ForeColor为字体颜色, BackGroundColor为背景颜色
void SetPos(int x, int y); //设置光标位置, x为行坐标位置, y为列坐标位置
void GetPos(int &x, int &y); //获取当前控制台光标位置, x为行坐标位置, y为列坐标位置

```

● 主要函数调用关系图

