

数据结构与算法

课程设计报告

(2020—2021学年 第二学期)

系 别 信息与控制工程

专 业 计算机科学与技术

班 级 计算机2001班

学 号 203230133

姓 名 颜奇昌

指导教师 贺飞

完成时间

评定成绩

目录

**第一章 简介**1

**第二章 需求分析**1

2.1 功能分析1

2.2 功能模块结构图2

**第三章 概要设计**3

  3.1 数据类型的定义3

3.2 数据结构ER图4

**第四章 详细设计**4

4.1 流程图4

* 1. 功能描述5
     1. 功能页面5
     2. 基本信息查看：5
     3. 可供查询的地点及其基本信息介绍6
     4. 路径的查询14
     5. 最短路径的查询15
     6. 寻找某个地点出发的行径16
     7. 路径的扩充17

4.2.8 路径的撤销18

4.2.9 新地图的建立19

4.2.10 退出系统20

**第五章 系统测试**21

5.1 简介22

5.2 开发软件23

5.3 测试结果24

**第六章 总结**25

1. 简介

城市公交是专门服务于市民出行的客运企业。它是城市社会和经济活动的重要组成部分。伴随着国民经济和城市建设的快速发展,城市经济的繁荣,人口的增加,城市必须解决好人们出行的需求城市公交接关系着城市的经济发展和居民生活,对城市经济具有全局性、先导的影响,城市公交以其方便、快捷、容量大而成为城市交通的主体。

第二章 需求分析

2.1功能分析

 本系统中的数据来源于标准输入设备（如键盘）和文件，可以实现对交通 图城市、 根据需要可查询某城市两点之间的最短 距离，以及路径。本系统要实现的功能有：可供查询的地点及其基本信息介绍,可供查询的地点及其基本信息介绍，查询两点间最短路径,查询从某点出发的最短连通路径，路径的扩充,路径的撤销,新建整张地图具体如下：

1.添加交通图信息 能录入新数据（地区和路径）。当录入了重复的地区和路径时，则提示数据录入重复并取消录入；当交通图中超过15个地区时，存储空间已满，不能再录入新数据；录入的新数据能按递增的顺序自动进行条目编号。

2.查询单源最短路径能够实现输入起点名后，查询出其到各点的最短路径，输出该起点到的其他终点的最短路径。

3.查询多源最短路径输入起点名和终点名，查询出两个点的最短路径，并输出该最短路径。

4.更新交通图信息根据给定的地区名能够修改该地区的名字。或者输入两个地区，修改一条路径的距离。

  5.删除交通图信息根据输入的地区名，删除与该城市有关的所有路径。输入两个地区可以删除一条路径。

 6.读取、保存信息能够实现退出系统时把交通图中的信息保存在一个文件中，在程序瑕疵运行时能够读取出来。

7. 扩建一个新的整体的地图。

8.退出系统。

2.2功能模块结构图

城市区域查询系统

查询功能

修改功能

新建图

退出系统

删除地点

增加地点

查删除路径

增添路径

查询路线问题

查询名称及详情

查询全部信息

第三章 概要设计

 3.1 数据类型的定义

定义交通图城市的元素类

typedef struct\_city{ char name[10];城市名

 struct \_path \* firstpath；//第一个路径

}AdjList[15],CityNode;

定义交通图的路径元素类型

typedef struct \_path! int adjcity/邻接点域

int distance；//距离

 struct \_path \*nextpath//下一个路径

 }

PathNode,\*PathPtr;

定义交通图类型

 typedef struct { int cities;

int paths;

AdjiList list;

 }Graph；

全局变量

typedef struct node;

ypedef struct vertexnode;

typedef struct adjlist;

typedef struct queue;

typedef struct mst\_point;

3.2 数据结构ER图

构成

图graph

地址 name

第四章 详细设计

4.1 流程图

城市区域查询系统

删除地点

路径的撤销

新建整张地图 (未完成)

径路径的扩充

查询从某点出发的最短连通路径(未完成)

查询两点间最短路径(未完成)

查询任意两点间所有的路径(未完成)

地图的基本信息

新增地点

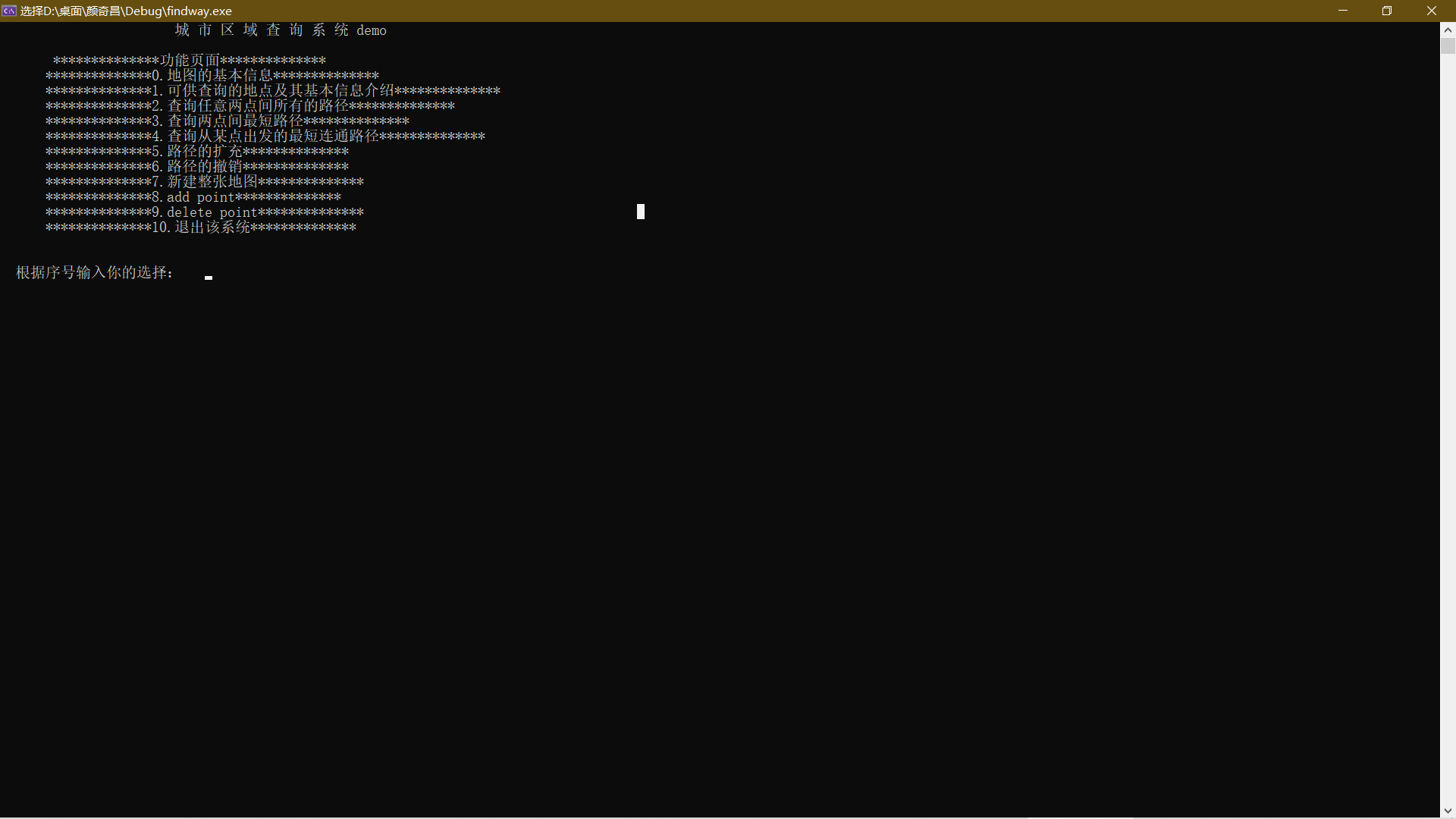
可供查询的地点及其基本信息介绍

退出系统

4.2 功能描述

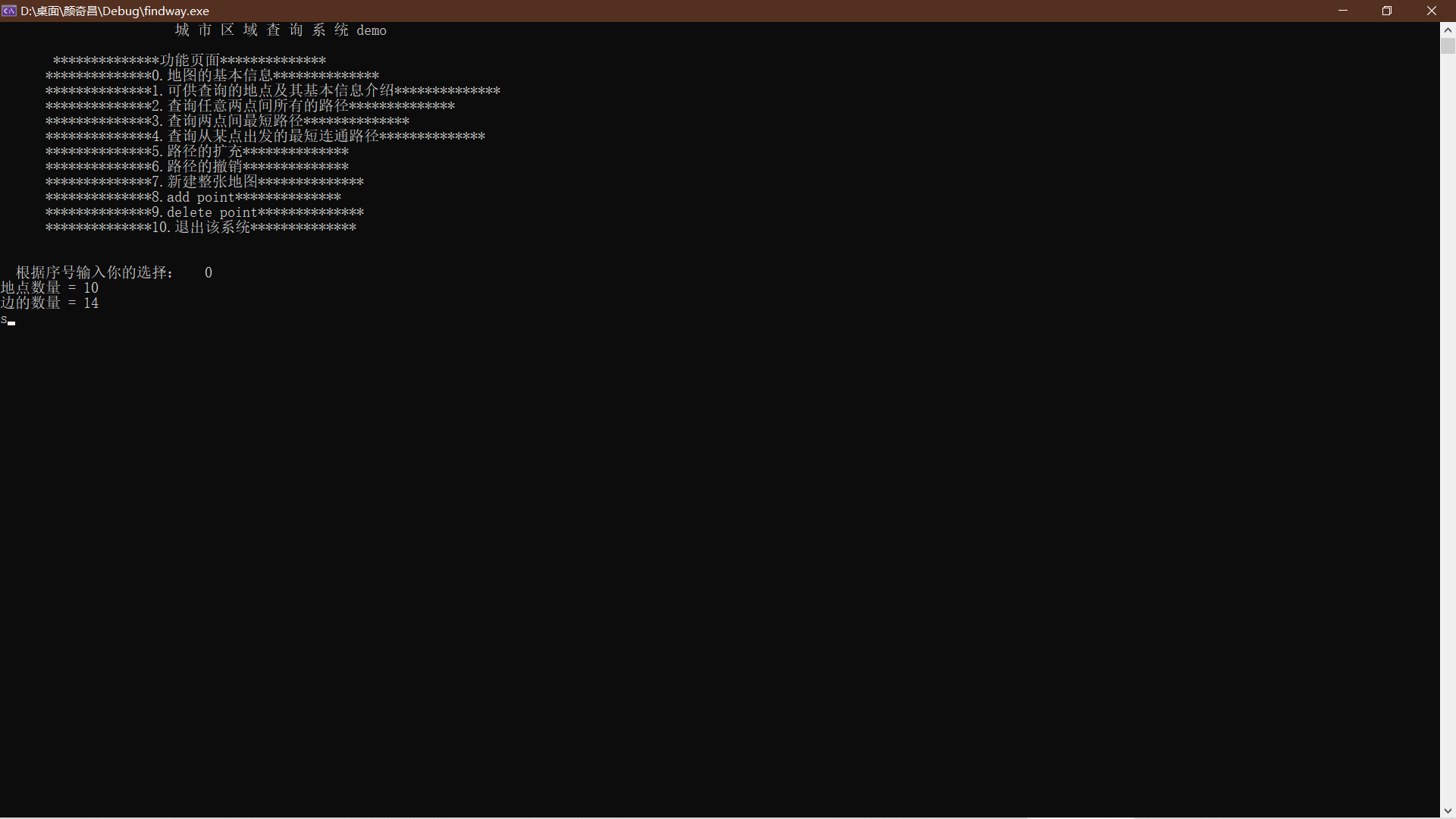
4.2.1 功能页面：

程序主页面，共有10个选择项目，用户可根据自己的需求选择执行序列。



4.2.2 基本信息查看：

选择序号“0”，按回车健，就可以显示地图基本信息的查看，共有10个地点，边数量为14。



4.2.3 可供查询的地点及其基本信息介绍：

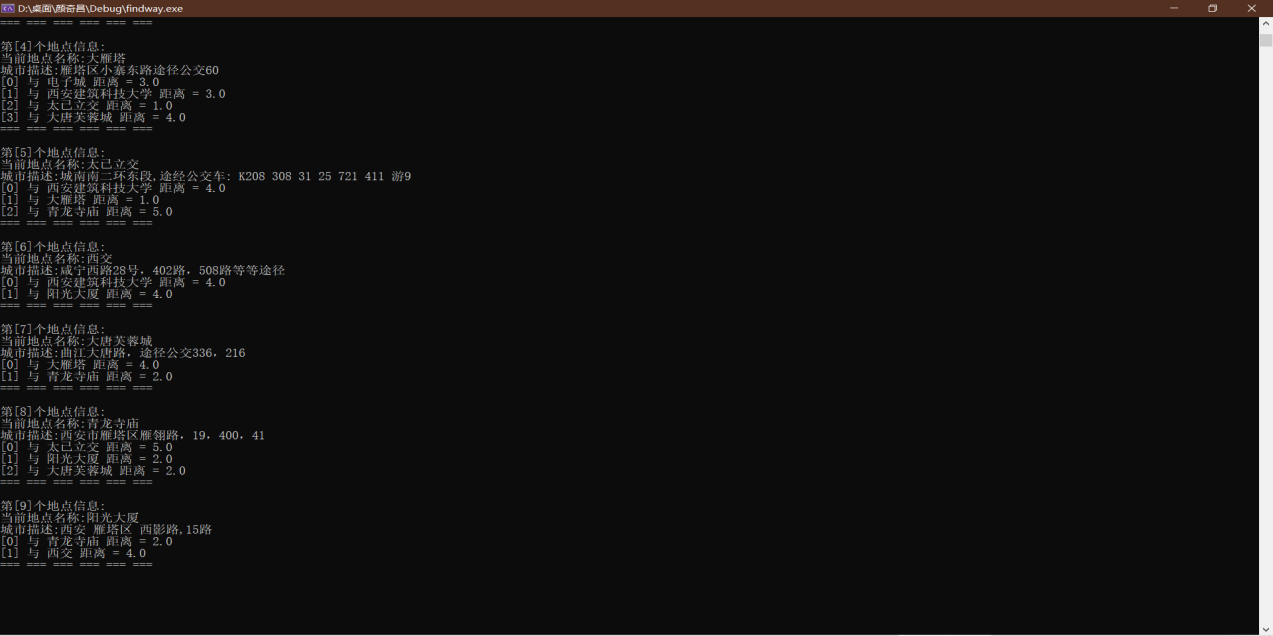
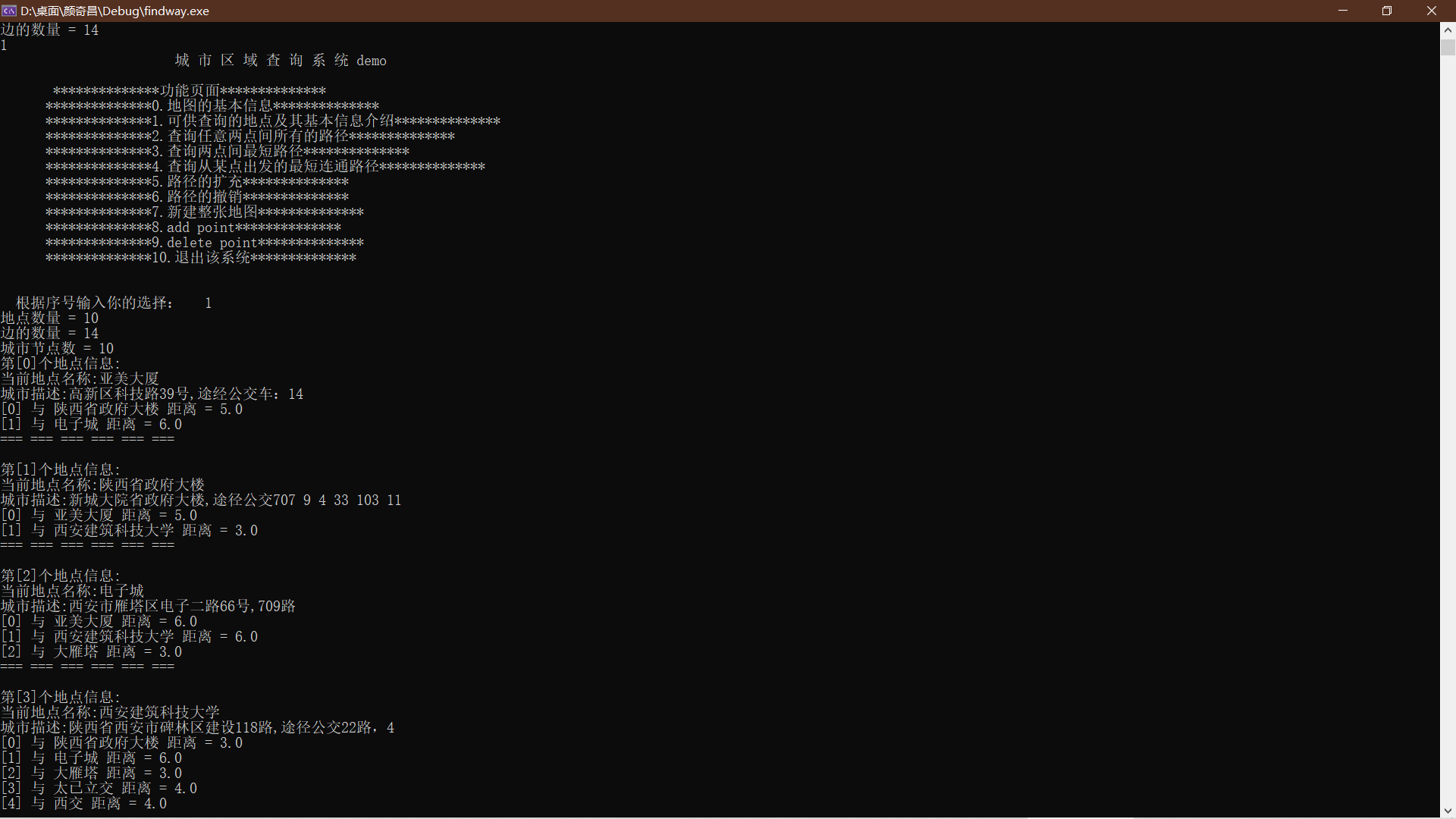
选择序号1，然后按回车键便显示如下界面，图中显示了10个地点，以及其基本信息。比如：当前地点名称:亚美大厦

城市描述:高新区科技路39号,途经公交车：14

[0] 与 陕西省政府大楼 距离 = 5.0

[1] 与 电子城 距离 = 6.0

方便用户更快，更准确的找到自己所需的地点，以及一些介绍，以便用户参考。

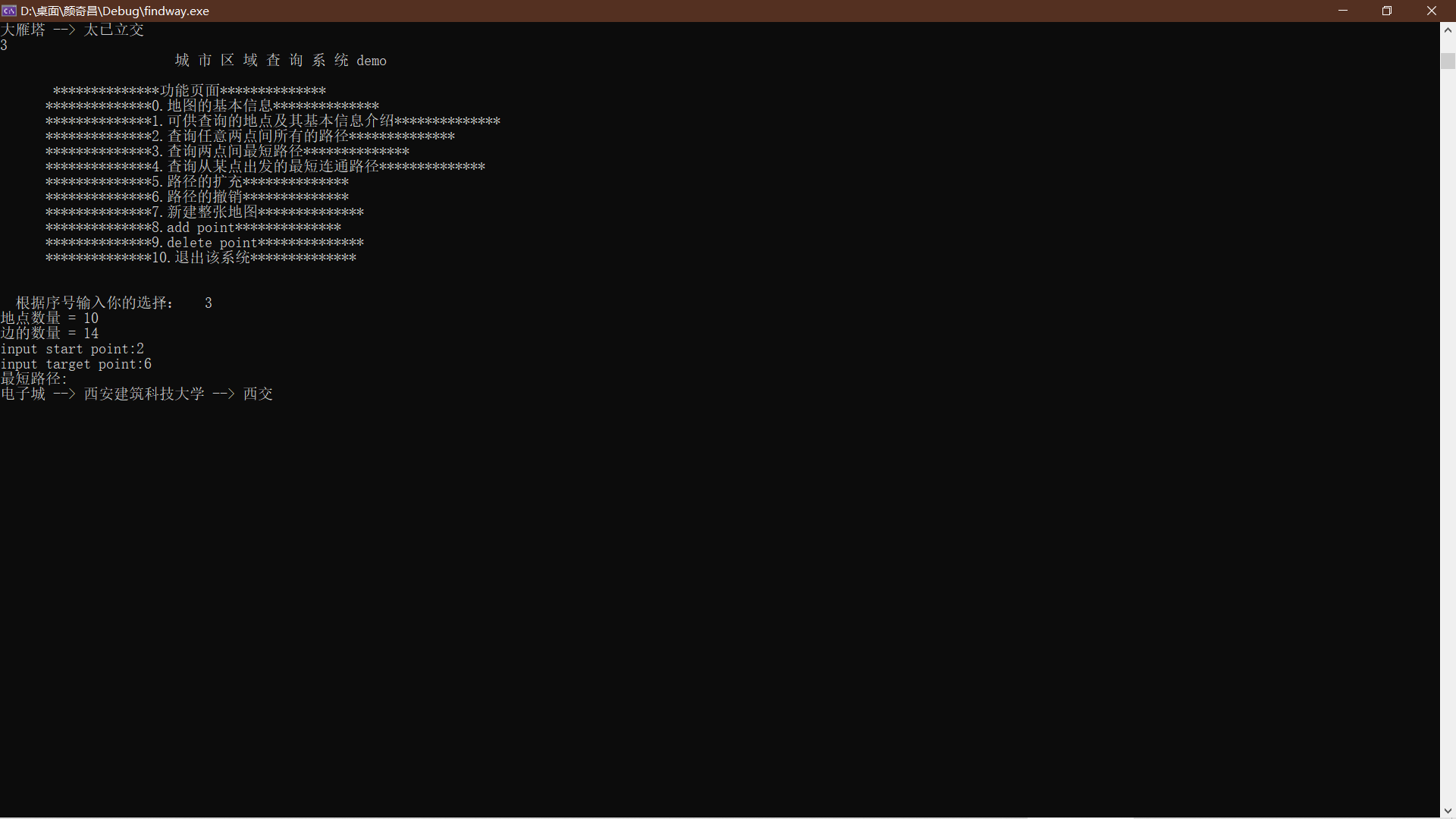


4.2.4 路径的查询：

选择序号“2”，按回车健便显示如下界面，然后从10个地点中任意选择两个地点，便可显示该两点的路径。用户可以根据自己的位置，选择附近的交通站，根据自己的目的地，选择不同的交通路线。比如：我选择的是“1”，“2”，两个地点。便有如下：

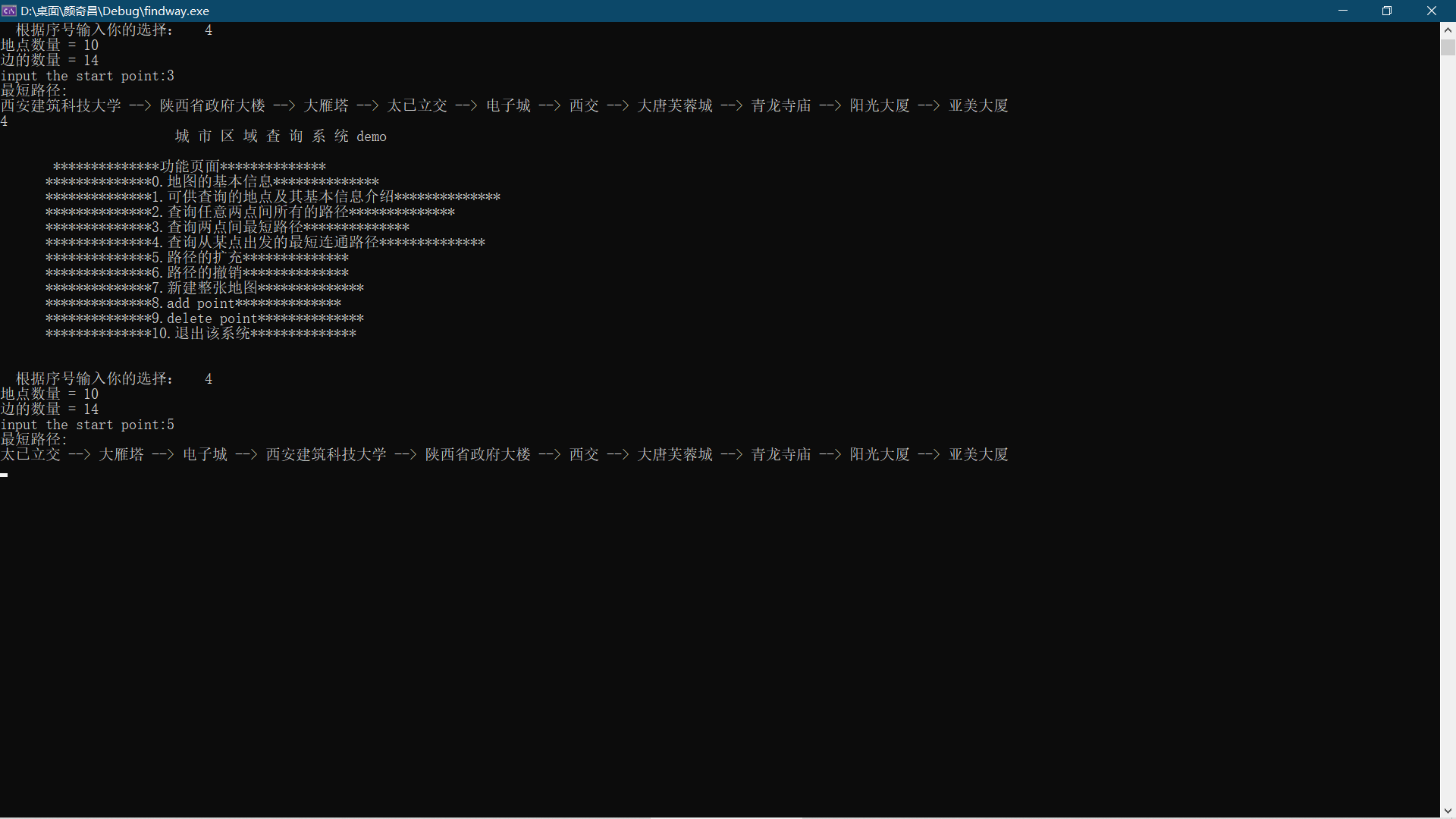
4.2.5 最短路径的查询：

选择序号“3”，按下回车键，然后从10个地点中选择任意两个地点查看它们的最短路径，以达到节约用户时间的目的。比如我选择操作的是“2”，“6”地点就会有如下界面显示：

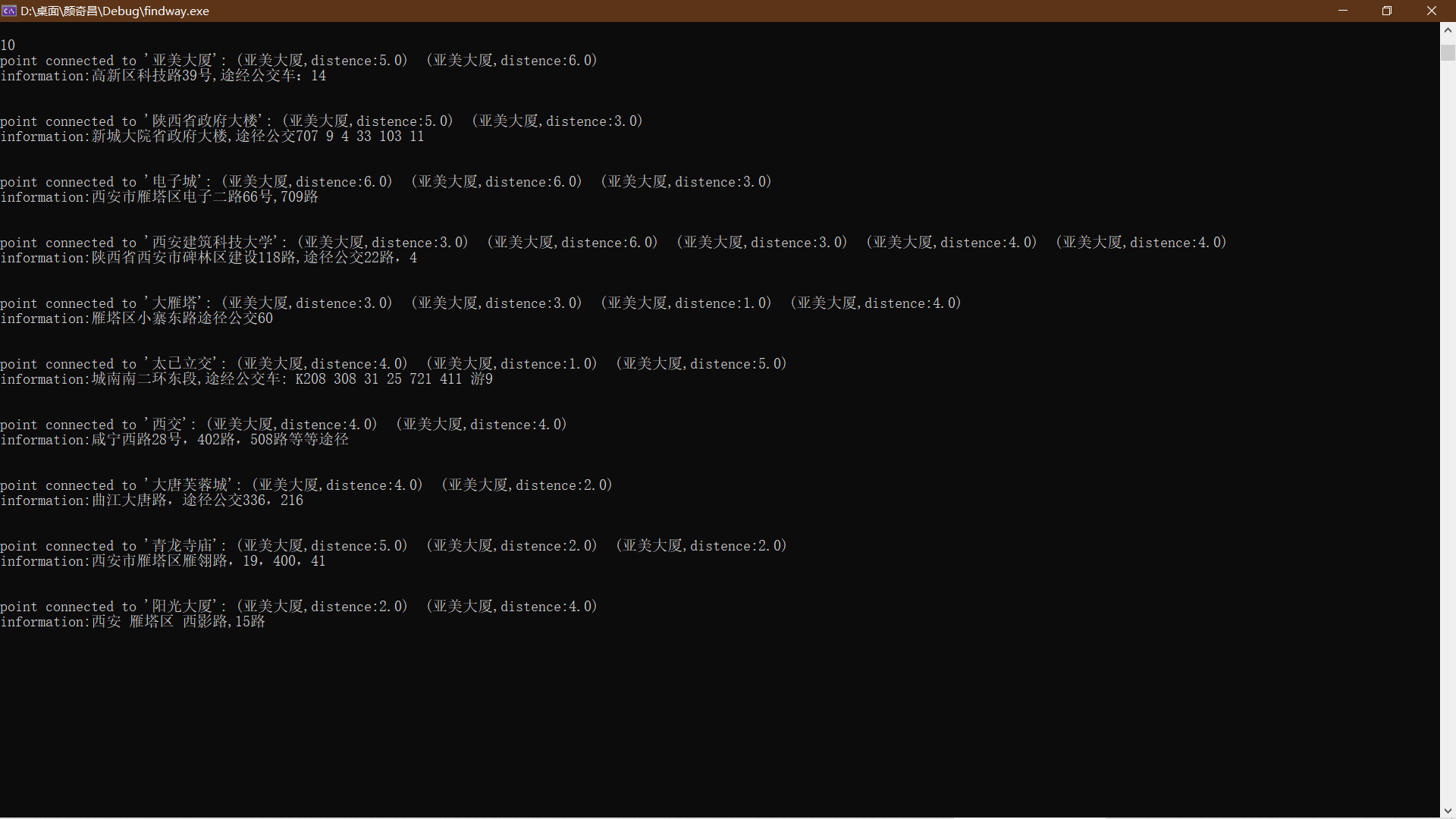
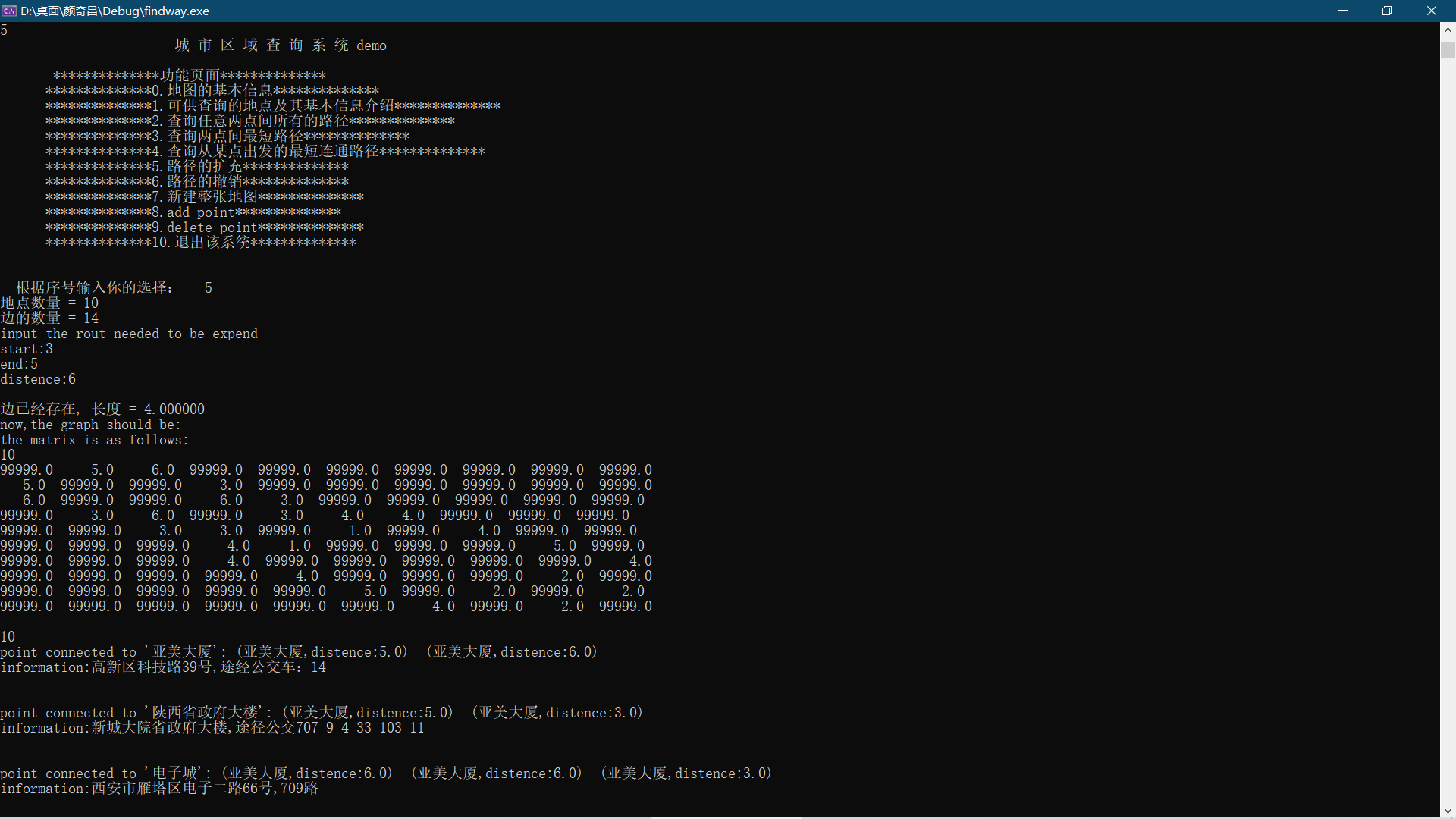


4.2.6 寻找某个地点出发的行径：

选择序号“4”，按回车键，就会显示出如下界面，用户可以从10个地点选择，自己从某地点出发，选择最短的路径，离目的地。比如我选择的是“4号”地点，就会如下图所示，给出路径。

4.2.7 路径的扩充：  
对于地图路径的扩充，是利用的数据结构中的二叉树，最主要是扩充外部路径长度，和内部路径长度。

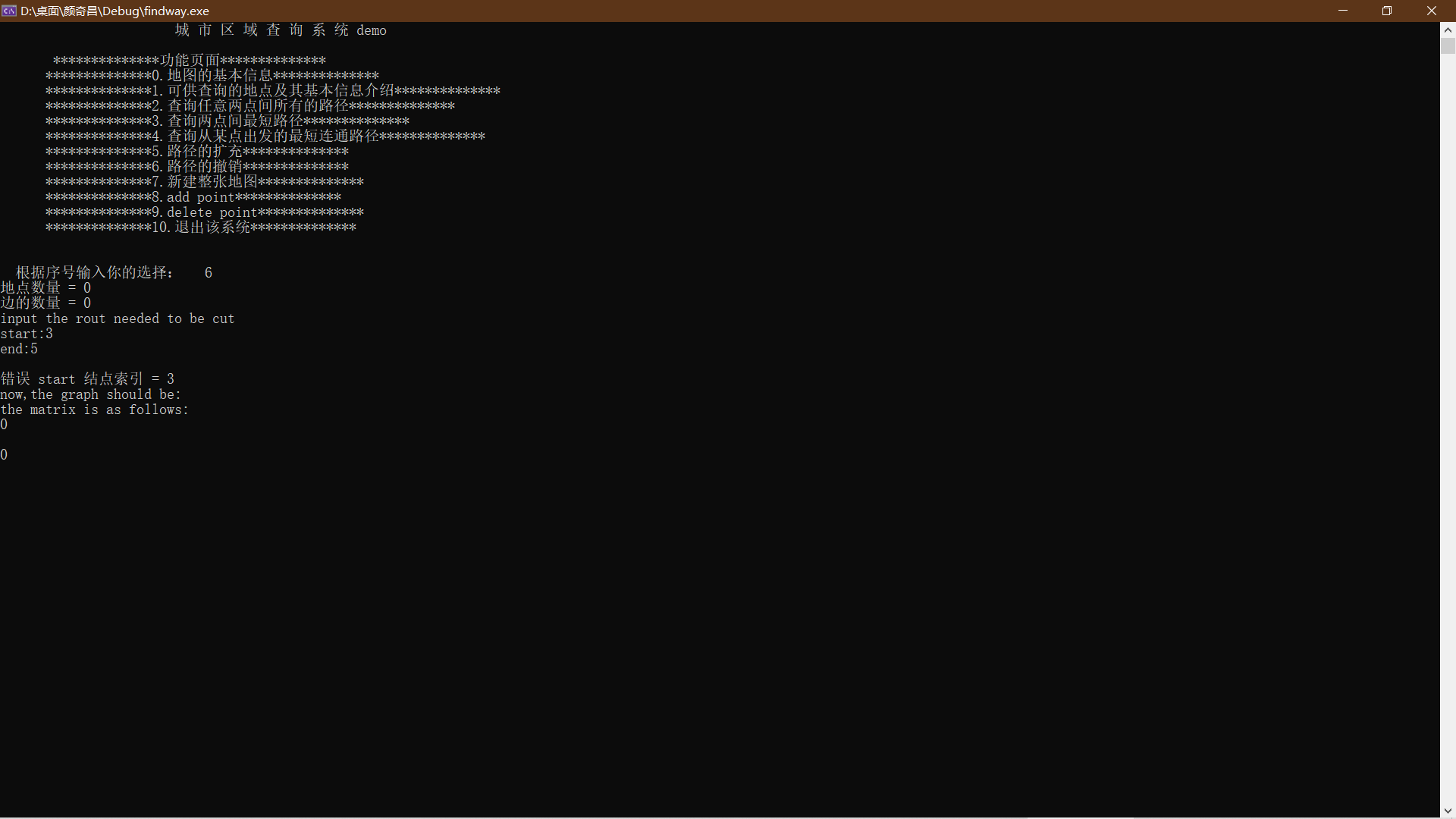
操作如下，选择序号“5”，然后按下回车键，进行路径的扩充。



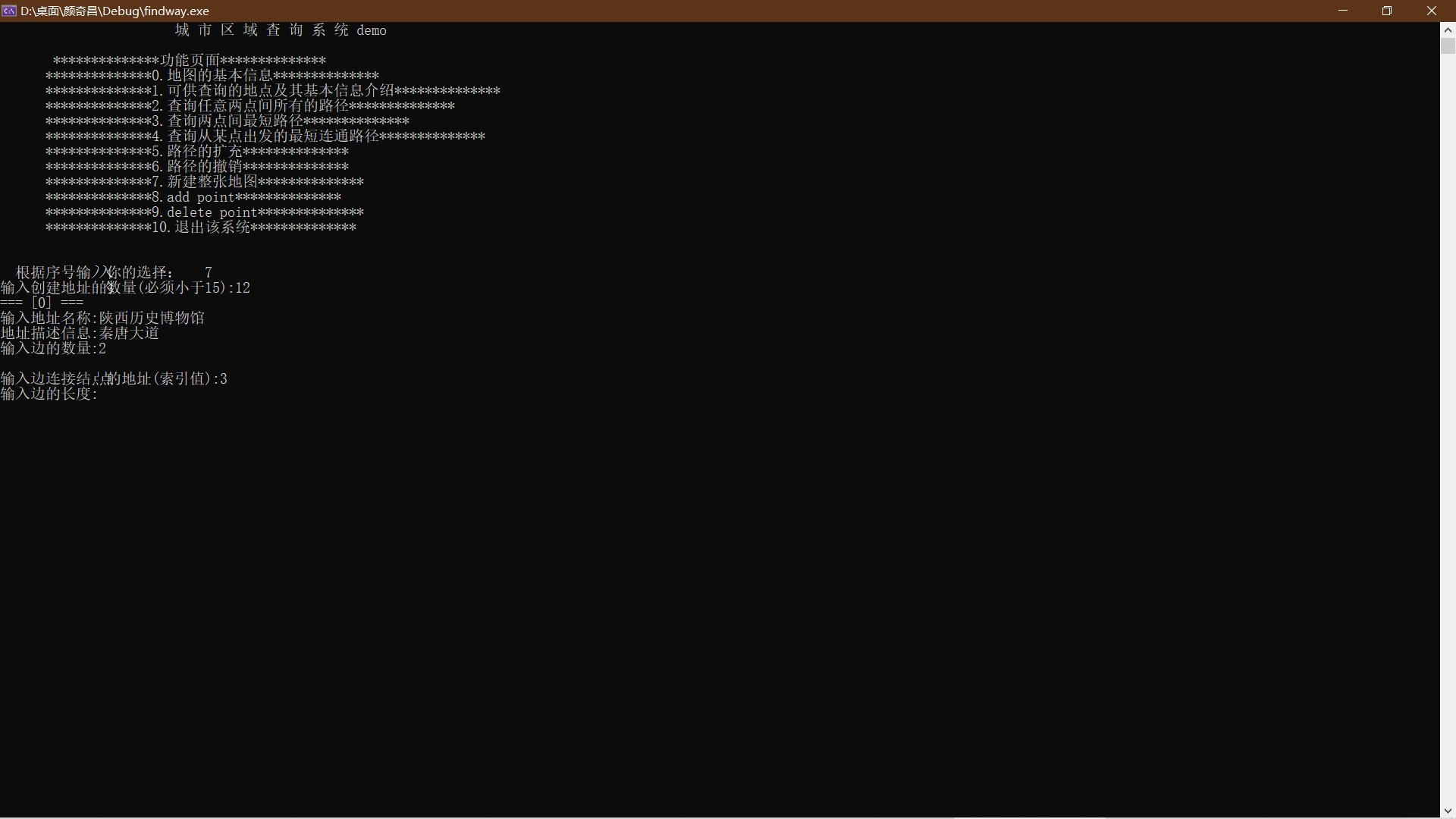
4.2.8 路径的撤销：

和路径的扩充是同一个原理。

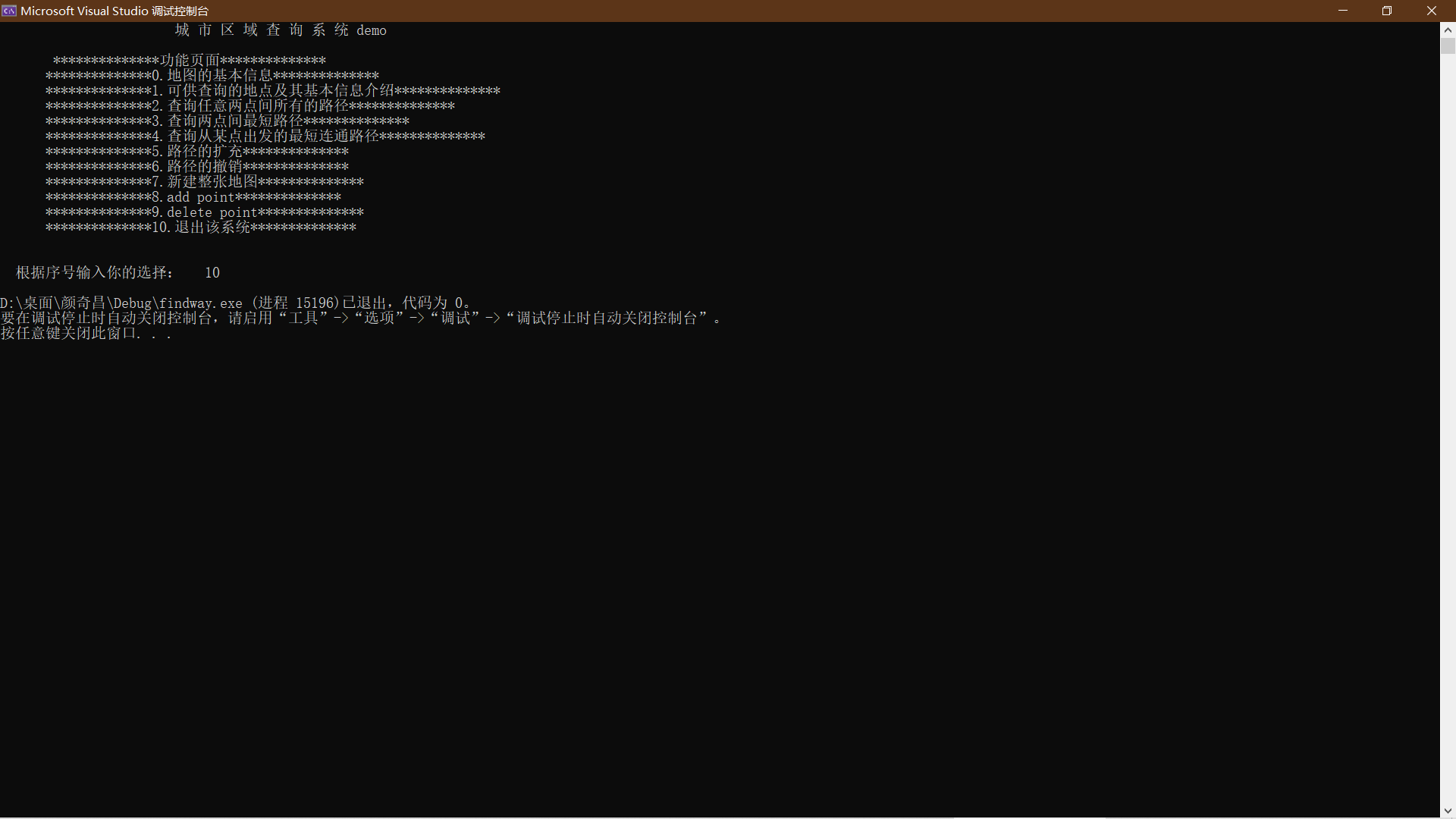
如图上图所示，我扩充的路径是“3”和“5”，然后选择序号“6”，进行路径的撤销，就会如下图所示。

4.2.9 新地图的建立

选择序号“7”，按回车键，便可自行新建地图。比如我想新建个陕西历史博物馆的地址，便可自行输入“陕西历史博物馆”，然后填充一些基本信息即可。就如下图所示：

4.2.9 新地图的建立

选择序号：“10”，按下回车键即可安全退出系统。

 第五章 系统测试

5.1 简介

城市区域查询系统.

5.2 开发软件

Visual Studio（简称VS或MSVS）是微软公司推出的开发工具套件系列产品，是目前最流行的 Windows平台应用程序开发环境，其包括了整个软件生命周期中所需要的大部分工具，如UML工具、代码管控工具、集成开发环境等。 Visual Studio可以用来创建Windows平台下的Windows应用程序和网络应用程序，也可以用来创建网络服务、智能设备应用程序和Office插件，还可开发安卓平台应用及IOS平台应用。

5.3 测试结果

易操作性：系统操作简单易用。

易浏览性：系统界面组件提示操作信息，方便指导用户操作。

可靠性：存在错误录入时给出相应的提示信息。

第六章 总结

在最初进行这个作业时，我当时就是个小白，不知道从哪里开始。主要问题是，自己编的代码不能运行，因为自己的基础知识没学好，编写的程序不能运行，没搞清楚函数与指针以及最基本的C语言语句还有一些数据结构的内容也不是特别的属下，比如“二叉树的建立，哈夫曼树”等，我开始在书上查找，并且认真去看书，然后一步步的进行修改自己的代码，一直到成功。自己的基础知识不够完善，就应该回头去学习，复习学过的知识，然后运用在实践操作中、比如这次作业。主动的去弥补自己的不足。

开头编写程序，没有头绪不知从何写起，需要慢慢去思考，编写什么系统，如何去写，具体分为几个步骤，然后还要去选择C语言的那些知识。不花费时间去慢慢思考根本不会有任何思路，然后我会去慢慢思考，上网查找自来资料，去借鉴模仿别人的，毕竟是第一次把C语言知识运用于实践中，借鉴他们的思路和基本模式，然后自己再去思考更改，最后形成了自己的步骤和思路，然后慢慢去编写程序。

第一次写的程序编码特别乱，自己都看不清晰，然后就要去一个一个的改正，让它看起来工整，如果出现错误以方便自己去改正，以后编写程序要思绪清晰，才能写出一个正确的程序，还看起来舒服，工整。

经过这次课程设计作业我学到的东西有很多，一是数据结构方面的知识；二是算法方面的知识；三是编程思想的升华；四是解决问题的思路更加明晰。数据结构方面，我学会了灵活运用链表和数组，并学会运用了将二者合二为一的邻接表。邻接表是一个常见的图的存储结构。然而它也有很多方面的不足。如：寻找特定的弧结点，需要遍历弧链表，比较麻烦。因此在操作图的结构方面 出现了很多的问题， 我耐心得针对每个问题，想出每一种不同的解决方法。虽然邻接表在操作上很复杂，但是便于理解，在构造一个又一个函数之后，各种基本操作也变得简单了。尤其是书上迪杰斯特拉算法的C语言描述，用的是邻接矩阵，而我的又是无向图，所以非常难以理解。好在我不断的琢磨算法，不断的改造，以适应我的程序和存储结构，最终完成了算法的实现，令我十分欣喜。鼓舞 了我学习算法和编程的信心。 算法方面，通过改造迪杰斯特拉算法， 我了解了算法并不是固定不变的，只要了解算法的精髓，便可更好利用。编程思想和解决问题方面，我学会了耐心，学会了理性的用逻辑图的方式解决问题。更加端正了我解决问题态度，以及学会了解决问题的套路。