

重庆师范大学

《数据结构课程设计》

课程名称: 数据结构

题 目: 校园地图导航

学 院: 计算机与信息科学学院

专业年级: 计科(3+2)

小组组长: 梁人义

小组成员: 母淑婷

指导教师: 张万里 职称: 讲师

2021 年 12 月 20 日

目录

一、需求分析.....	3
二、问题描述.....	3
三、基本要求.....	3
四、概要设计.....	4
1、算法思路.....	4
2、工作分配.....	6
3、程序模块.....	6
五、主程序.....	7
六、运行截图.....	25
七、总结.....	37

校园地图导航

一、需求分析

在现在的生活中，很多大学的占地面积比较大，建筑比较多，很多大学生基本局限于教学楼，宿舍，食堂和超市这些地方，对于其他的办公楼都比较陌生。因此，在很多时候需要办理某些业务或者事情的时候都需要询问方向，这样很不方便。对于重庆师范大学（大学城校区），需要开发一个校园地图导航系统，用于帮助学生在校园里进行地图导航。

相信这个校园地图导航系统能够解决学生对校园的环境感到陌生，迷失方向的问题。有了这个校园地图导航系统，学生不需要再为找不到地点而烦恼，也不用再向其他人不停问路，这样可以提高学生办事效率，节约大量时间。

二、问题描述

校园地图导航系统覆盖了重庆师范大学大学城校区，通过平面图的方式展现校园风貌，为广大师生提供校园内各种设施（教学单位、生活设施、行政机构等）的位置查询及导航服务，方便新生和家长浏览校园各单位、院系及各类设施分布情况，能够更快更方便的熟悉校园环境，融入校园生活。

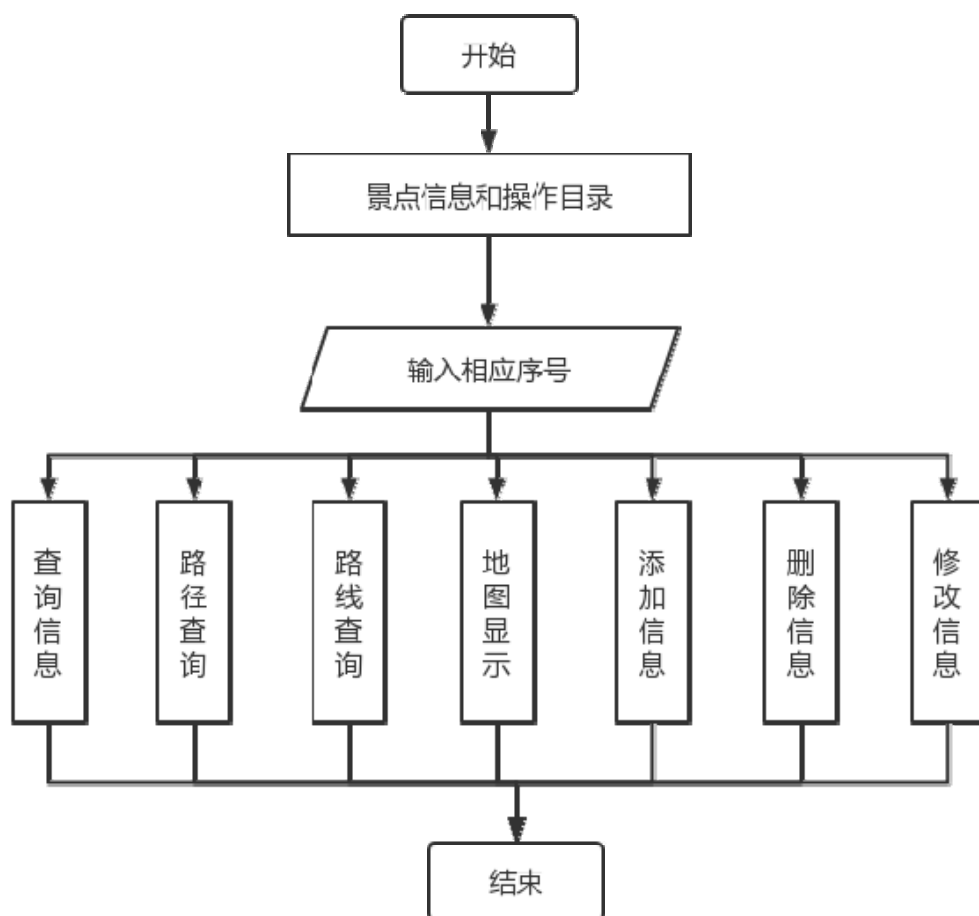
校园地图导航系统主要提供了校园地图和地图导航等功能。其中，校园地图：提供校内各机构（包括教学单位、行政机构等）、校内生活设施浏览服务；提供校内各个位置点之间的线路导航功能。为广大新生提供食堂点、宿舍点的定位和导航等路线信息。

三、基本要求

本次实验任务是设计一个简易的重庆师范大学的校园导航平面图，所含景点不少于 8 个。以图中顶点表示学校内各景点，存放景点的名称、景点介绍信息等；以边表示路径，存放路径长度信息。设计需要包括下列要求：

- （1） 导航系统的信息包括：景点信息查询、景点最短路径查询、校园地图显示查询、景点路径添加退出信息系统等。
- （2） 系统能实现的操作功能如下：

- ① 景点信息查询：至少实现两种方法进行景点查询；
- ② 景点最短路径查询：输入你现在所在位置，并且输入你想要去的地方，程序会计算出两地之间的最短距离，并且给出线路图
- ③ 校园地图显示查询：可以展现校园的平面地图；
- ④ 景点路径添加\删除：可以在已有的景点查询中进行景点路径的添加\删除；
- ⑤ 景点路径修改：可以修改已有的任一景点查询
- ⑥ 退出信息系统：可以退出校园地图导航系统



四、概要设计

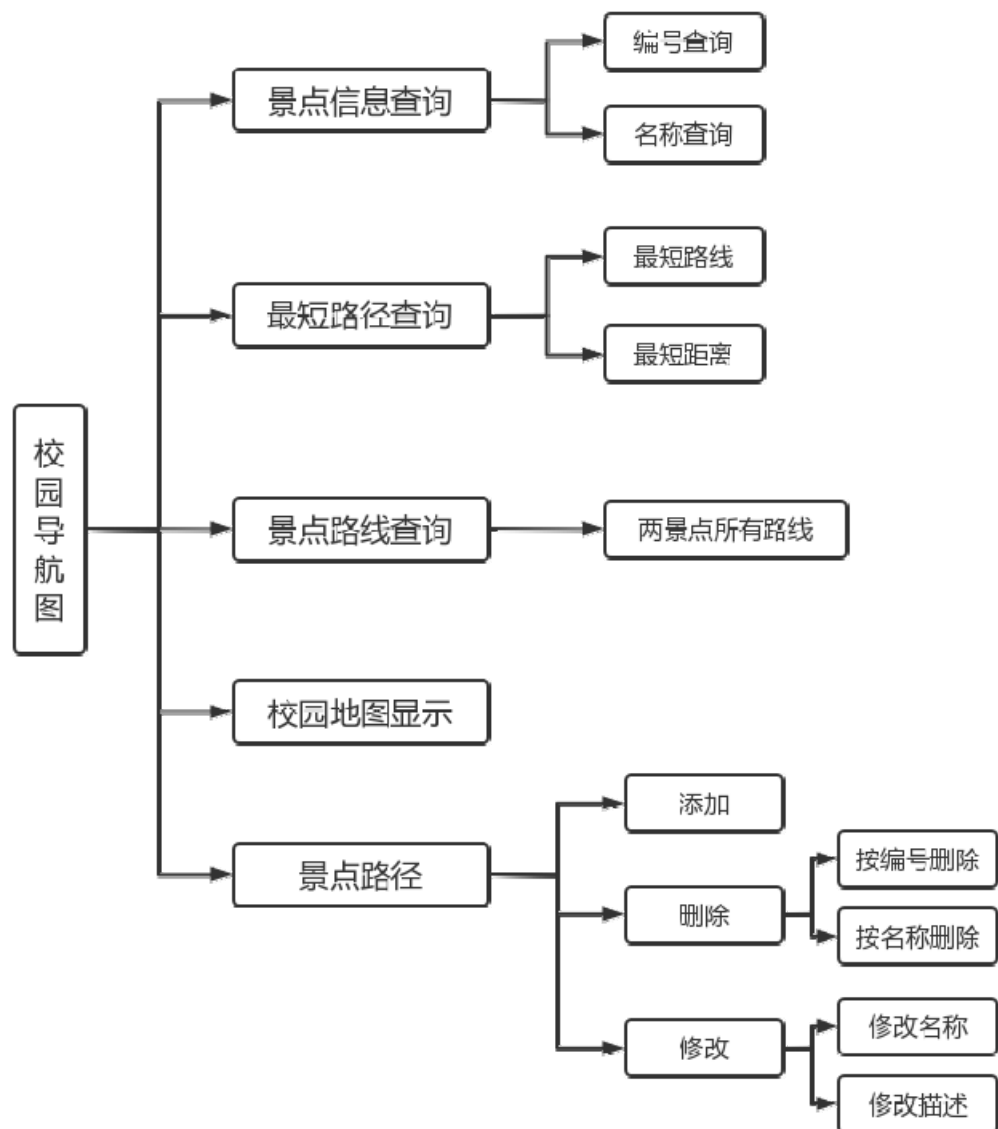
1、算法思路

用 C++ 进行编写，使用了结点结构体，邻接矩阵和邻接表，迪杰斯特拉算法最短路径函数，还包括整个框架的构思，图、栈、队列等等的运用，最终构成了整个程序。

(1) 进入导航系统，遍历所有景点及查询两景点间的最短路径都是通过迪杰斯特拉来实现最短路算法，将各景点间的权值存入二维数组 `map[]` 中，通过迪杰斯特拉算法，能显示出两景点间的最短路径，在通过这个点去更新别的边的权值，同时将更新的点存储，然后放入栈中输出即为到终点所走路径。

(2) 查询所有景点信息是先将景点信息存储然后输出，景点信息是利用 `strcpy()` 函数将各景点名称及信息存入到结构体中。查询那个输出对应下标存储的字符串。

(3) 定义其他各个板块函数，然后返回主界面直接调用函数。



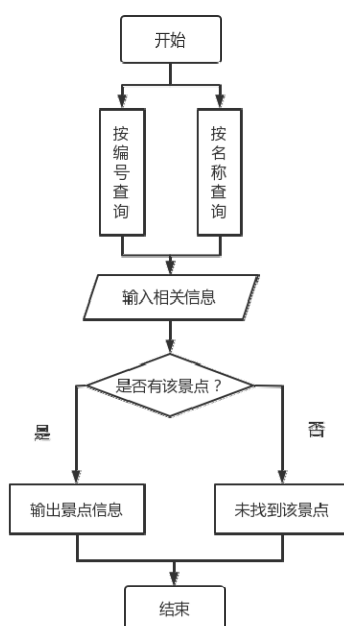
2、工作分配

- (1) 框架设计： 由母淑婷进行构思，同时参考其他人的意见。
- (2) 系统设计： 由母淑婷进行框架的编写。
- (3) 程序设计： 由梁人义进行各项功能的编辑。
- (4) 程序调试： 由梁人义进行程序的调试与调整。
- (5) 文档制作，由梁人义、母淑婷共同完成文档的编辑工作。

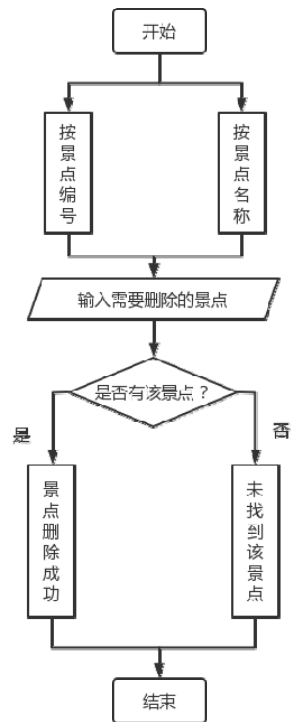
3、程序模块

本校园地图导航系统主要包括以下功能：

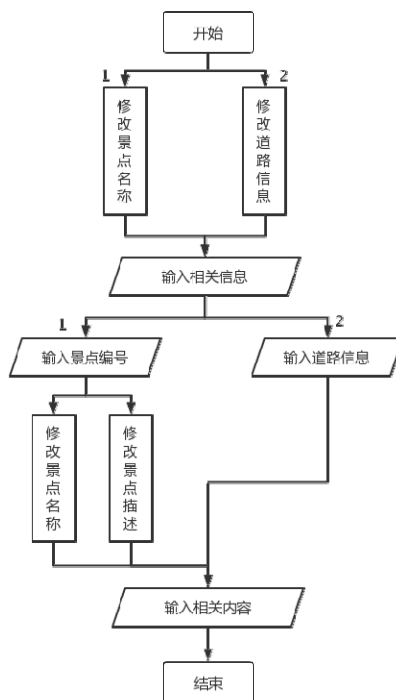
- ① 景点信息查询：提供校园任意景点相关信息的介绍用两种方法实现，一种以景点编号查询，一种以景点名字查询；下面是它的流程图：



- ② 景点最短路径查询：输入你现在所在位置，并且输入你想要去的地方，程序会计算出两地之间的最短距离，并且给出线路图；
- ③ 两景点的路径查询：能够提供所查询两景点的所有能达路径；
- ④ 校园地图显示查询：可以展现校园的平面地图；
- ⑤ 景点路径添加：可以在已有的景点查询中进行景点路径的添加；
- ⑥ 退出信息删除：可以删除已有的任一景点查询；下面是它的流程图：



⑦ 景点路径修改：可以修改已有的任一景点查询；下面是它的流程图：



⑧ 退出信息系统：可以退出校园地图导航系统

五、主程序

程序 1. 设计 homePage() 函数为系统主菜单

```
void homePage(){
int cmd; //用户选择的编号
int v0,v1;
while(1){
    printf("\n\n");
    printf("\t ┌───────────────────────────────────┐ \n");
    printf("\t │ ┌──────────────────────────────────────────┐ │ \n");
    printf("\t │ │  欢 迎 使 用 校 园 地 图 导 航 系 统  │ │ \n");
    printf("\t │ └──────────────────────────────────────────┘ │ \n");
    printf("\t │ ***** │ \n");
    printf("\t │ *                *                * │ \n");
    printf("\t │ * 1.景点信息查询      *  2.景点最短路径      * │ \n");
    printf("\t │ *                *                * │ \n");
    printf("\t │ ***** │ \n");
    printf("\t │ *                *                * │ \n");
    printf("\t │ * 3.两景点的路径      *  4.校园地图显示      * │ \n");
    printf("\t │ *                *                * │ \n");
    printf("\t │ ***** │ \n");
    printf("\t │ *                *                * │ \n");
    printf("\t │ * 5.景点路径添加      *  6.退出信息删除      * │ \n");
    printf("\t │ *                *                * │ \n");
    printf("\t │ ***** │ \n");
    printf("\t │ *                *                * │ \n");
    printf("\t │ * 7.景点路径修改      *  8.退出信息系统      * │ \n");
    printf("\t │ *                *                * │ \n");
    printf("\t │ ***** │ \n");
    printf("\t └───────────────────────────────────┘ \n");
    printf("\t 请选择功能 1 - 8 :");
```


[illegible]

```
void search(){ /* 查询景点信息 */
    int num;

    int i;

    char c;

    char name[20];

    do{
        system("cls");

        c=SearchMenu();

        switch (c){
            case 'l':

                system("cls");

                printf("\n\n\t\t 请输入您要查找的景点编号: ");

                scanf("%d",&num);
```

```

        for(i=0;i<NUM;i++){
            if(num==MG.vex[i].number){
                printf("\n\n\t\t\t 您要查找景点信息如下:");
                printf("\n\n\t\t\t 景点名称: %-25s\n",MG.vex[i].name);
                printf("\n\n\t\t\t 景点介绍: %-25s\n\n",MG.vex[i].description);
                printf("\n\t\t\t 按任意键返回...");
                getchar();
                getchar();
                break;
            }
        }
        if(i==NUM){
            printf("\n\n\t\t\t 没有找到! ");
            printf("\n\n\t\t\t 按任意键返回...");
            getchar();
            getchar();
            }break;
    case '2':
        system("cls");
        printf("\n\n\t\t\t 请输入您要查找的景点名称: ");
        scanf("%s",name);
        for(i=0;i<NUM;i++){
            if(!strcmp(name,MG.vex[i].name)){
                printf("\n\n\t\t\t 您要查找景点信息如下:");
                printf("\n\n\t\t\t 景点名称: %-25s\n",MG.vex[i].name);
                printf("\n\n\t\t\t 景点介绍: %-25s\n\n",MG.vex[i].description);
                printf("\n\t\t\t 按任意键返回...");
                getchar();
                getchar();break;
            }
        }

```

```

    }
    if(i==NUM){
        printf("\n\n\t\t\t 没有找到! ");
        printf("\n\n\t\t\t 按任意键返回...");
        getchar();
        getchar();
        }break;
    }
}while(c!='3');
}

```

程序 4. 设计 CreateUDN()函数为造图函数，地图初始化

```

void CreateUDN(int v,int a){ //造图函数，给景点初始化
    int i,j;
    MG.vexnum=v;    //初始化结构体中的景点数
    MG.arcnum=a; //初始化结构体的边数
    for(i=0;i<20;++i) MG.vex[i].number=i; //初始化每一个景点的编号
    strcpy(MG.vex[0].name,"学校正门");
    strcpy(MG.vex[0].description,"位于重庆市沙坪坝区大学城中路 37 号，交通便利。");
    strcpy(MG.vex[1].name,"弘德楼");
    strcpy( MG.vex[1].description,"弘德楼作为我校的主教学楼，共有 139 间教室，可容 11800 位学子入座。");
    strcpy(MG.vex[2].name,"校友会堂");
    strcpy(MG.vex[2].description,"这是一个让广大校友回到母校聚集一堂，这里寄托了重师就是我的家的美好心愿。");
    strcpy(MG.vex[3].name,"足球场");
    strcpy(MG.vex[3].description,"这是重师学子强身健体的地方，丰富课余生活。");
}

```

```

strcpy(MG.vex[4].name,"一食堂");
strcpy(MG.vex[4].description,"这是重师学子最喜欢的一食堂。");
strcpy(MG.vex[5].name,"图书馆");
strcpy(MG.vex[5].description,"历史悠久，文化积淀。供同学安静学习的环境。
");
strcpy(MG.vex[6].name,"三食堂");
strcpy(MG.vex[6].description,"这是重师学子最朴素的三食堂。");
strcpy(MG.vex[7].name,"雅风苑");
strcpy(MG.vex[7].description,"这是女生宿舍，离南门最近，离宏德楼最远的
宿舍。");
strcpy(MG.vex[8].name,"美术学院");
strcpy(MG.vex[8].description,"这是重师学美术学子上课的地方。");
//这里把所有的边假定为 20000，含义是这两个景点之间是不可到达 */
for(i=0;i<20;++i)
    for(j=0;j<20;++j)
        MG.view[i][j].adj=Max;
/*下边是可直接到达的景点间的距离，由于两个景点间距离是互相的，所以
要对图中对称的边同时赋值。*/
MG.view[0][1].adj=MG.view[1][0].adj=50;
MG.view[1][3].adj=MG.view[3][1].adj=70;
MG.view[0][6].adj=MG.view[3][0].adj=250;
MG.view[1][4].adj=MG.view[4][1].adj=80;
MG.view[2][4].adj=MG.view[4][2].adj=100;
MG.view[3][5].adj=MG.view[5][3].adj=90;
MG.view[5][2].adj=MG.view[2][5].adj=100;
MG.view[4][6].adj=MG.view[6][4].adj=75;
MG.view[4][7].adj=MG.view[7][4].adj=300;
MG.view[2][7].adj=MG.view[7][2].adj=400;
MG.view[7][8].adj=MG.view[8][7].adj=40;
}

```

程序 5. 设计 narrate() 函数为说明函数，显示校园的景点

```
void narrate(){ //说明函数，介绍每个景点
    int i,k=0;
    printf("\n\t*****欢迎使用重庆师范大学校园导游程序*****\n");
    printf("\t_____ \n");
    printf("\t 景点名称\t\t\t 景点描述\t\t\t\n");
    printf("\t_____ | _____ \n");
    for(i=0;i<NUM;i++){
        printf("\t(%2d )%-10s\t\t| %-8s\n",i,MG.vex[i].name,MG.vex[i].description);
//输出景点列表
        k=k+1;
    }
    printf("\t_____ | _____ \n");
}
```

程序 6. 设计 shortRoad() 函数为最短路径函数

```
void shortRoad(int num){ //迪杰斯特拉算法最短路径函数 num 为入口点的编号
    int v,w,i,t; // i、w 和 v 为计数变量
    int final[20];
    int min;
    for(v=0;v<NUM;v++){
        final[v]=0; //假设从顶点 num 到顶点 v 没有最短路径
        D[v]=MG.view[num][v].adj; //将与之相关的权值放入 D 中存放
        for(w=0;w<NUM;w++){ //设置为空路径
            P[v][w]=0;
            if(D[v]<20000){ //存在路径
                P[v][num]=1; //存在标志置为一
            }
        }
    }
}
```

```

        P[v][v]=1;           //自身到自身
    }
}
D[num]=0;
final[num]=1;           //初始化 num 顶点属于 S 集合
/* 开始主循环，每一次求得 num 到某个顶点的最短路径，并将其加入到 S
集合 */
for(i=0;i<NUM;++i){      // 其余 MG.vexnum-1 个顶点
    min=Max;              //当前所知离顶点 num 的最近距离
    for(w=0;w<NUM;++w)
        if(!final[w])    //w 顶点在 v-s 中
            if(D[w]<min){  //w 顶点离 num 顶点更近
                v=w;
                min=D[w];
            }
    final[v]=1;           // 离 num 顶点更近的 v 加入到 s 集合
    for(w=0;w<NUM;++w)    //更新当前最短路径极其距离
        if(!final[w]&&((min+MG.view[v][w].adj)<D[w])){ //不在 s 集合，并且
比以前所找到的路径都短就更新当前路径
            D[w]=min+MG.view[v][w].adj;
            for(t=0;t<NUM;t++)
                P[w][t]=P[v][t];
            P[w][w]=1;
        }
    }
}
}

```

程序 7. 设计 output()函数为输出函数

```

void output(int sight1,int sight2){    //输出函数
    int a,b,c,d,q=0;

```

```

a=sight2;    /* 将景点二赋值给 a */
if(a!=sight1)    /* 如果景点二不和景点一输入重合，则进行... */
{
    printf("\n\t 从   %s   到   %s   的   最   短   路   径   是
",MG.vex[sight1].name,MG.vex[sight2].name);/* 输出提示信息 */
    printf("\t(最短距离为 %dm.)\n\n\t",D[a]); /* 输出 sight1 到 sight2 的最
短路径长度，存放在 D[]数组中 */
    printf("\t%s",MG.vex[sight1].name);    /* 输出景点一的名称 */
    d=sight1;    /* 将景点一的编号赋值给 d */
    for(c=0;c<NUM;++c)
    {
        gate;;    /* 标号，可以作为 goto 语句跳转的位置 */
        P[a][sight1]=0;
        for(b=0;b<NUM;b++)
        {
            if(MG.view[d][b].adj<20000&&P[a][b]) /* 如果景点一和它的一个
临界点之间存在路径且最短路径 */
            {
                printf("-->%s",MG.vex[b].name); /* 输出此节点的名称 */
                q=q+1;    /* 计数变量加一，满 8 控制输出时的换行 */
                P[a][b]=0;
                d=b;    /* 将 b 作为出发点进行下一次循环输出，如此
反复 */

                if(q%8==0) printf("\n");
                goto gate;
            }
        }
    }
}
}
}
}

```

程序 8. 设计 Searchpath1()函数为查询函数，查询两个景点间的所有路径

```
void Searchpath1(MGraph g)/*查询两个景点间的所有路径*/
{
    int l=0;
    int k=0;
    int i,j;
    printf("选择出发景点: ");
    scanf("%d",&i);
    printf("选择目的地景点: ");
    scanf("%d",&j);
    for(;k<g.vexnum;k++)/*g.vexnumber 表示网中的顶点个数*/
        if(i==g.vex[k].number) i=k;/*在网中找到其编号与输入的出发景点的编号相同的顶点*/
    for(;l<g.vexnum;l++)
        if(j==g.vex[l].number) j=l;/*在网中找到其编号与输入的目的地景点的编号相同的顶点*/
    printf("\n 从 %s 到 %s 的所有游览路径有:\n\n",g.vex[i].name,g.vex[j].name);/*输出出发景点和目的地景点的名称*/
    disppath(g,i,j);/*调用 disppath 函数,用来输出两个景点间的所有路径*/
}
```

程序 9. 设计 Addnewsight()函数为添加函数，添加新的景点和路径

```
int Addnewsight(int n)
{
    int i;
    char name[100],description[1000];
    int length;
```



```

printf("请输入新景点的名称: \n");
scanf("%s",&name);
printf("请输入新景点的相关信息: \n");
scanf("%s",&description);
strcpy(MG.vex[n].name,name);
strcpy(MG.vex[n].description,description);
MG.vex[n].number=n;
for(i=0;i<n;i++)
{
    system("cls");
    narrate();
    printf("请输入此景点到第%d 个景点的距离 (单位: m) (同一景点或
不可到达用 20000 表示): \n",i);
    scanf("%d",&length);
    if(length!=20000)
        MG.arcnum++;
    MG.view[n][i].adj=MG.view[i][n].adj=length;
}
n++;
MG.vexnum++;
return n;
}

```

程序 10. 设计 Deletesight()函数为删除函数, 删除景点和路径

```

int Deletesight(int n)
{
    int i;
    int j;
    char c;
    int num;
    char name[20];

```

```

system("cls");
c=SearchMenu();
switch (c)
{
case '1':
    system("cls");
    narrate();
    printf("\n\n\t\t 请输入您要删除景点的编号: ");
    scanf("%d",&num);
    for(i=0;i<NUM;i++)
    {
        if(num==MG.vex[i].number)
        {
            for(j=0;j<NUM;j++)
            {
                if(MG.view[i][j].adj!=20000)
                    MG.arcnum--;
                MG.view[i][j].adj=MG.view[j][i].adj=20000;
            }
        }
    }
    for(;num<NUM;num++)
    {
        strcpy(MG.vex[num].name,MG.vex[num+1].name);

strcpy(MG.vex[num].description,MG.vex[num+1].description);

        }
        n--;

```

```

        printf("\n\t\t\t 按任意键返回...");
        getchar();
        getchar();
        break;
    if(i==NUM)
    {
        printf("\n\n\t\t\t 没有找到! ");
        printf("\n\n\t\t\t 按任意键返回...");
        getchar();
        getchar();
    }
    break;
case '2':
    system("cls");
    narrate();
    printf("\n\n\t\t\t 请输入您要删除景点的名称: ");
    scanf("%s",name);
    for(i=0;i<NUM;i++)
    {
        if(!strcmp(name,MG.vex[i].name))
        {
            num=i;
            for(j=0;j<NUM;j++)
            {
                if(MG.view[i][j].adj!=20000)
                    MG.arcnum--;
                MG.view[i][j].adj=MG.view[j][i].adj=20000;
            }
        }
    }
}
}

```

```

for(;num<NUM;num++)
    {
        strcpy(MG.vex[num].name,MG.vex[num+1].name);
        strcpy(MG.vex[num].description,MG.vex[num+1].description);
    }
    n--;
    printf("\n\t\t\t 按任意键返回...");
    getchar();
    getchar();
    break;
if(i==NUM)
{
    printf("\n\n\t\t\t 没有找到! ");
    printf("\n\n\t\t\t 按任意键返回...");
    getchar();
    getchar();
}
break;
}

return n;
}

```

程序 11.设计 Changesight()函数为修改函数，修改景点和路径

```

void Changesight()
{
    int a,b,length;
    char name[100];
    char description[1000];
    char p;
    char q;

```

```

int i;

int num;

p=Changemenu();

switch(p)
{
case'1':
    system("cls");

    narrate();

    printf("\n\n\t\t 请输入您要修改的景点编号: ");

    scanf("%d",&num);

    for(i=0;i<NUM;i++)
    {
        if(MG.vex[i].number==num)
        {
            q=Sightmenu();

            if(q=='1')
            {
                printf("请输入修改后的景点名称: \n");

                scanf("%s",&name);

                strcpy(MG.vex[num].name,name);

                printf("\t\t\t\t 修改成功! \n");
            }
            else if(q=='2')
            {
                printf("请输入修改后的景点信息: \n");

                scanf("%s",&description);

                strcpy(MG.vex[num].description,description);

                printf("\t\t\t\t 修改成功! \n");
            }
            else if(q=='e')

```

```

                p=Changemenu();
            }
        }
        printf("\n\t\t 按任意键返回...");
        getchar();
        getchar();
        break;
    if(i==NUM)
    {
        printf("\n\n\t\t\t 没有找到! ");
        printf("\n\n\t\t\t 按任意键返回...");
        getchar();
        getchar();
    }
    break;
    case '2':
        printf("\t\t 请输入道路一侧的景点序号: ");
        scanf("%d",&a);
        printf("\t\t 请输入道路另一侧的景点序号: ");
        scanf("%d",&b);
        printf("\t\t 请输入修改后的道路长度: ");
        scanf("%d",&length);
        MG.view[a][b].adj=MG.view[b][a].adj=length;
        printf("\t\t\t\t 修改成功! \n");
        printf("\t\t");
        getchar();
        system("PAUSE");
        break;
    }
}

```

程序 12. 设计 Changemenu()函数为修改函数，修改路径或顶点的选择菜单

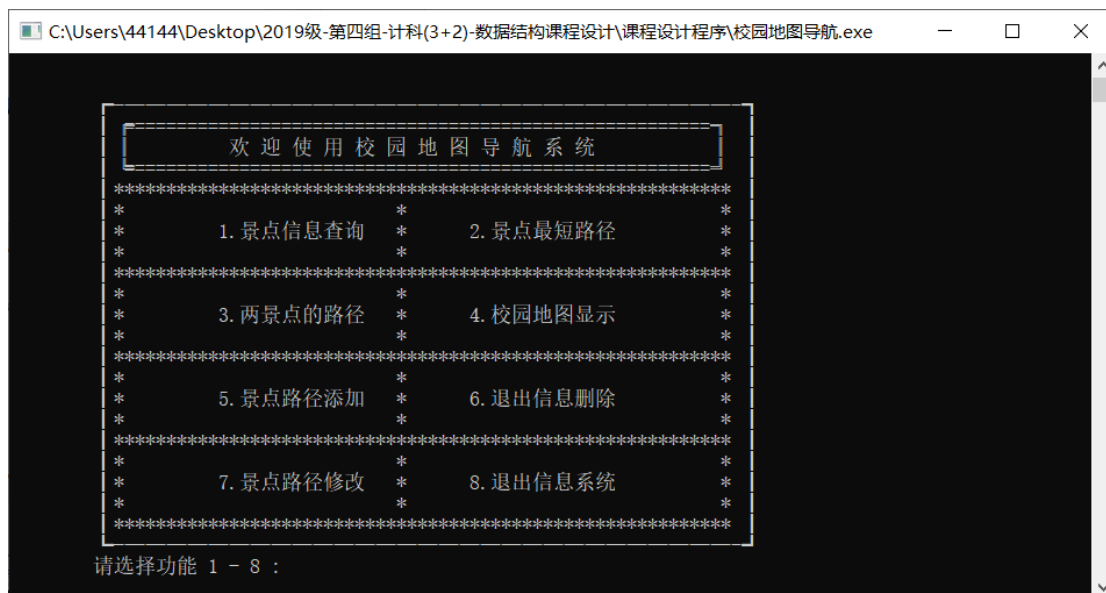
```
char Changemenu()
{
    char c;
    int flag;
    do{
        flag=1;
        system("cls");
        narrate();
        printf("\n\t\t\t┌──────────────────────────┐ \n");
        printf("\t\t\t\t| \n");
        printf("\t\t\t\t| 1、修改景点信息 | \n");
        printf("\t\t\t\t| 2、修改道路信息 | \n");
        printf("\t\t\t\t| e、返回 | \n");
        printf("\t\t\t\t| \n");
        printf("\t\t\t└──────────────────────────┘ \n");
        printf("\t\t\t\t请输入您的选择: ");
        scanf("%c",&c);
        if(c=='1' || c=='2' || c=='e')
            flag=0;
    }while(flag);
    return c;
}
```

程序 13. 设计 Sightmenu()函数为选择函数，选择需该景点的菜单

```
char Sightmenu()
{
    char c;
    int flag;
```


六、运行截图

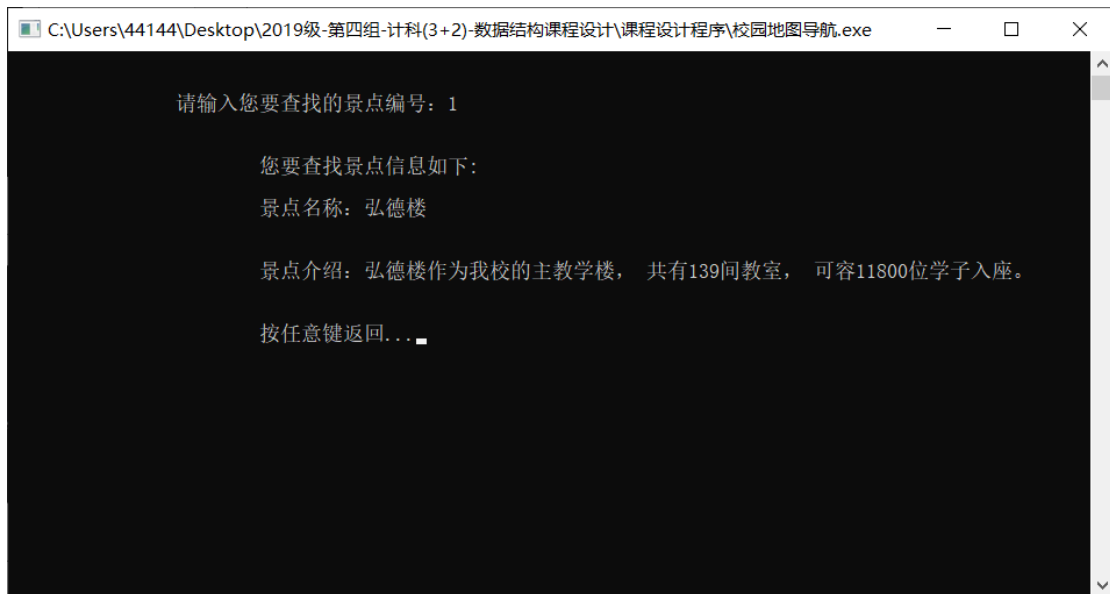
1、 点击编译运行出现校园导航程序系统主菜单：



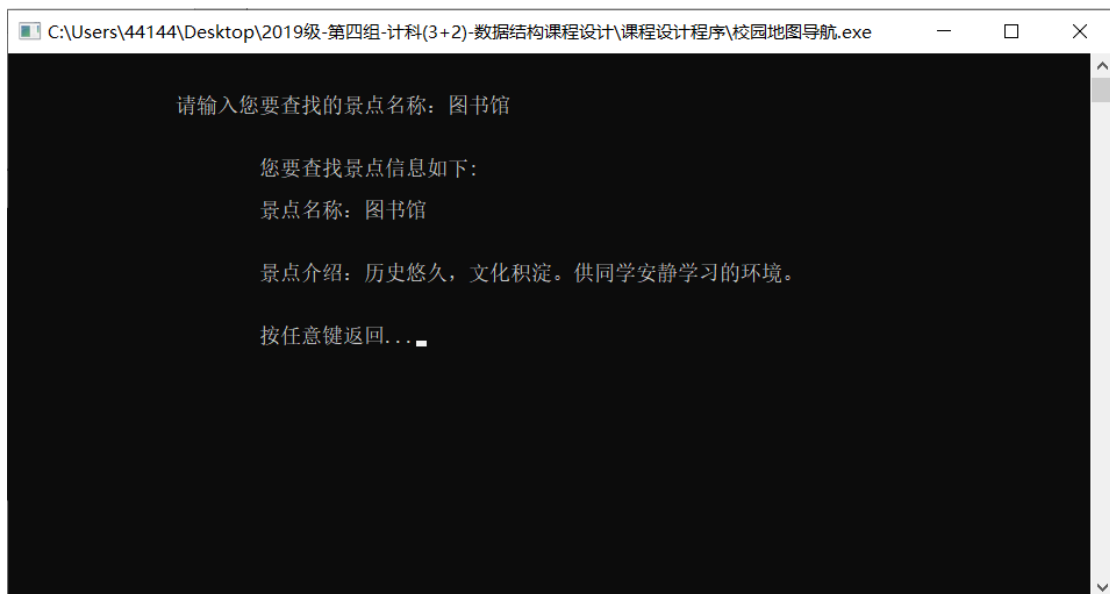
2、 选择主菜单中的“1”是景点信息查询，会出现按编号查询，景点名称查询，或是返回主菜单，选择你想要查询的方式即可：



选择“1”按照景点编号查询：

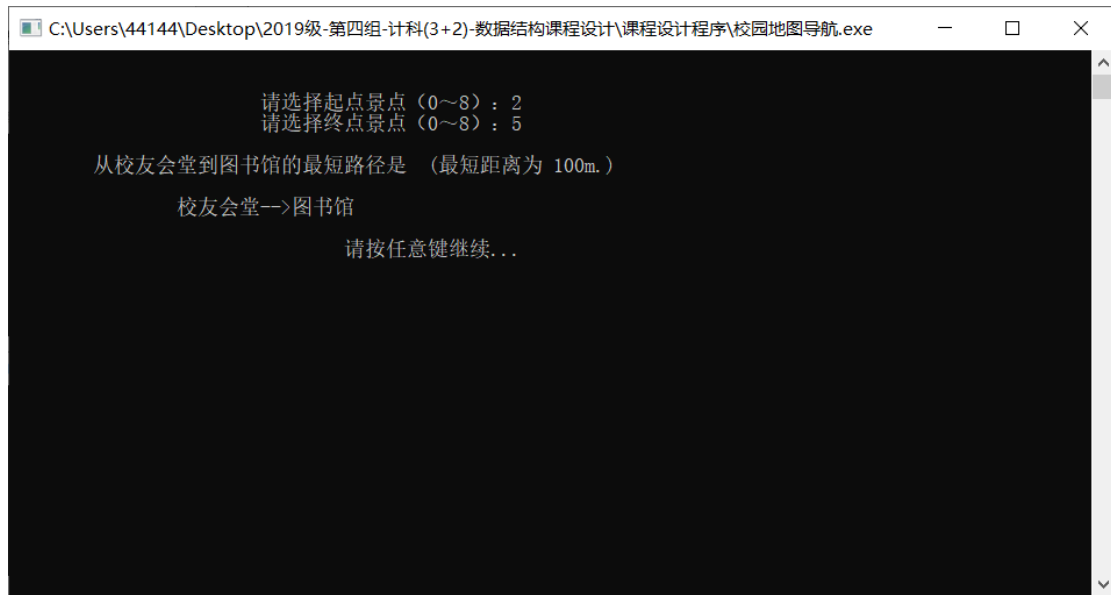


选择“2”按照景点名称查询:

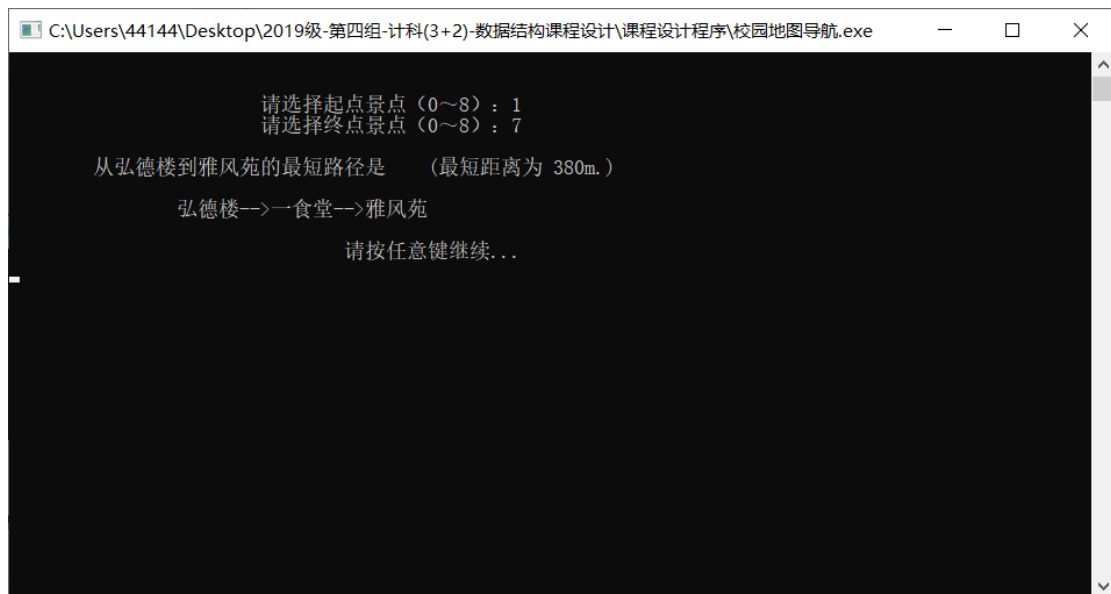


3、选择主菜单中的“2”是景点最短路径查询，选择起点和终点景点，会得

到最短路径（景点 2-景点 5）：



最短路径（景点 1-景点 7）：



4、选择主菜单中的“3”是两景点路径，选择出发景点和目的地景点，会得

到通往目的地的所有路线（景点 3-景点 4）:

选择C:\Users\44144\Desktop\2019级-第四组-计科(3+2)-数据结构课程设计\课程设计程序\校园地图导航.exe

*****欢迎使用重庆师范大学校园导游程序*****

景点名称	景点描述
(0) 学校正门	位于重庆市沙坪坝区大学城中路37号，交通便利。
(1) 弘德楼	弘德楼作为我校的主教学楼，共有139间教室，可容11800位学子入座。
(2) 校友会堂	这是一个让广大校友回到母校聚集一堂，这里寄托了重师就是我的家的美好心愿。
(3) 足球场	这是重师学子强身健体的地方，丰富课余生活。
(4) 一食堂	这是重师学子最喜欢的一食堂。
(5) 图书馆	历史悠久，文化积淀。供同学安静学习的环境。
(6) 三食堂	这是重师学子最朴素的三食堂。
(7) 雅风苑	这是女生宿舍，离南门最近，离弘德楼最远的宿舍。
(8) 美术学院	这是重师学美术学子上课的地方。

选择出发景点: 3
选择目的地景点: 4

从足球场到一食堂的所有游览路径有:

第1条: 足球场->学校正门->弘德楼->一食堂
第2条: 足球场->学校正门->三食堂->一食堂
第3条: 足球场->弘德楼->学校正门->三食堂->一食堂
第4条: 足球场->弘德楼->一食堂
第5条: 足球场->图书馆->校友会堂->一食堂
第6条: 足球场->图书馆->校友会堂->雅风苑->一食堂

请按任意键继续...

通往目的地的所有路线（景点 0-景点 7）

C:\Users\44144\Desktop\2019级-第四组-计科(3+2)-数据结构课程设计\课程设计程序\校园地图导航.exe

*****欢迎使用重庆师范大学校园导游程序*****

景点名称	景点描述
(0) 学校正门	位于重庆市沙坪坝区大学城中路37号，交通便利。
(1) 弘德楼	弘德楼作为我校的主教学楼，共有139间教室，可容11800位学子入座。
(2) 校友会堂	这是一个让广大校友回到母校聚集一堂，这里寄托了重师就是我的家的美好心愿。
(3) 足球场	这是重师学子强身健体的地方，丰富课余生活。
(4) 一食堂	这是重师学子最喜欢的一食堂。
(5) 图书馆	历史悠久，文化积淀。供同学安静学习的环境。
(6) 三食堂	这是重师学子最朴素的三食堂。
(7) 雅风苑	这是女生宿舍，离南门最近，离弘德楼最远的宿舍。
(8) 美术学院	这是重师学美术学子上课的地方。

选择出发景点: 0
选择目的地景点: 7

从学校正门到雅风苑的所有游览路径有:

第1条: 学校正门->弘德楼->足球场->图书馆->校友会堂->一食堂->雅风苑
第2条: 学校正门->弘德楼->足球场->图书馆->校友会堂->雅风苑
第3条: 学校正门->弘德楼->一食堂->校友会堂->雅风苑
第4条: 学校正门->弘德楼->一食堂->雅风苑
第5条: 学校正门->三食堂->一食堂->弘德楼->足球场->图书馆->校友会堂->雅风苑
第6条: 学校正门->三食堂->一食堂->校友会堂->雅风苑
第7条: 学校正门->三食堂->一食堂->雅风苑

请按任意键继续...

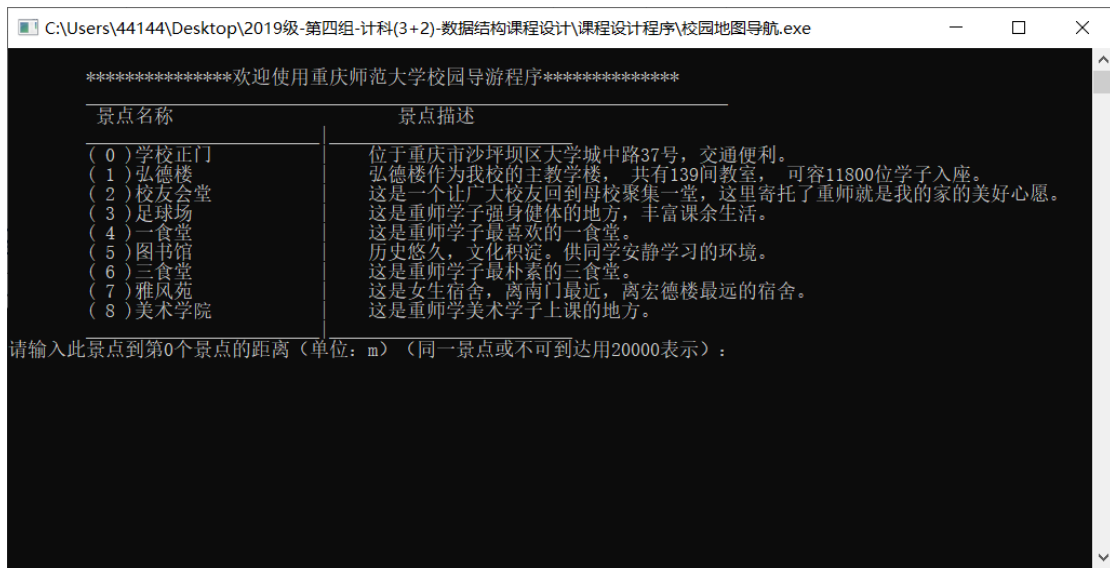
5、选择主菜单中的“4”是校园地图显示，会显示校园的平面地图



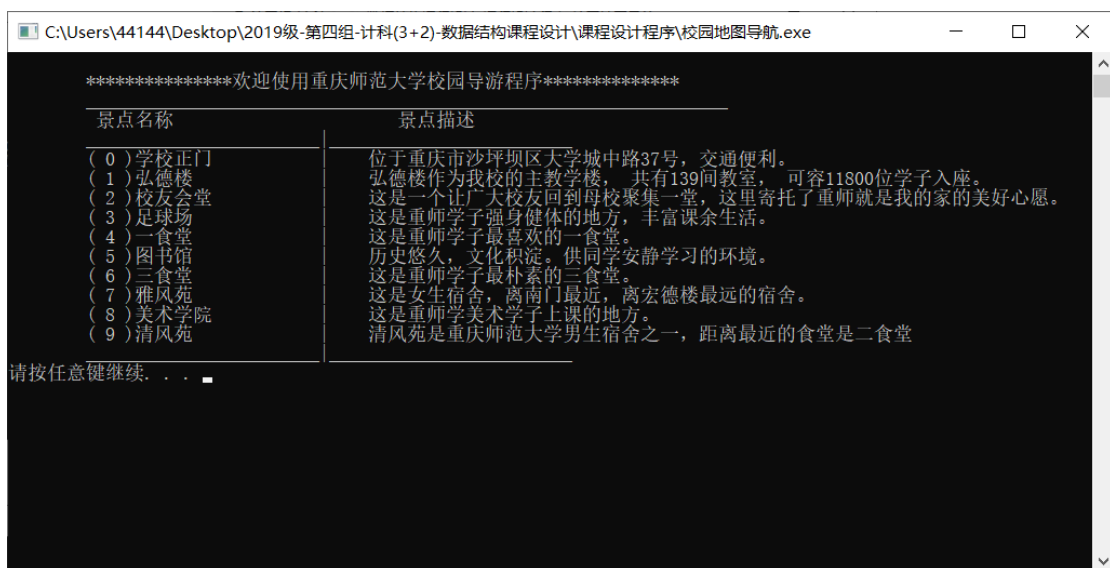
6、选择主菜单中的“5”是景点路径添加，输入景点名称和景点相关信息：



输入此景点到第 0 -8 个景点的距离：



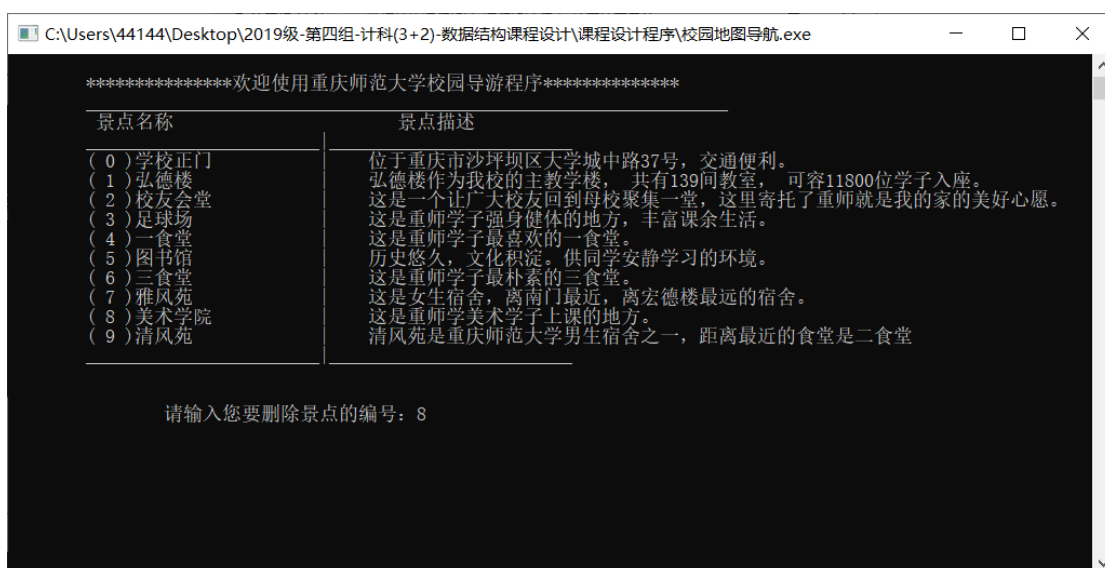
查询景点，已添加名称为清风苑的景点：



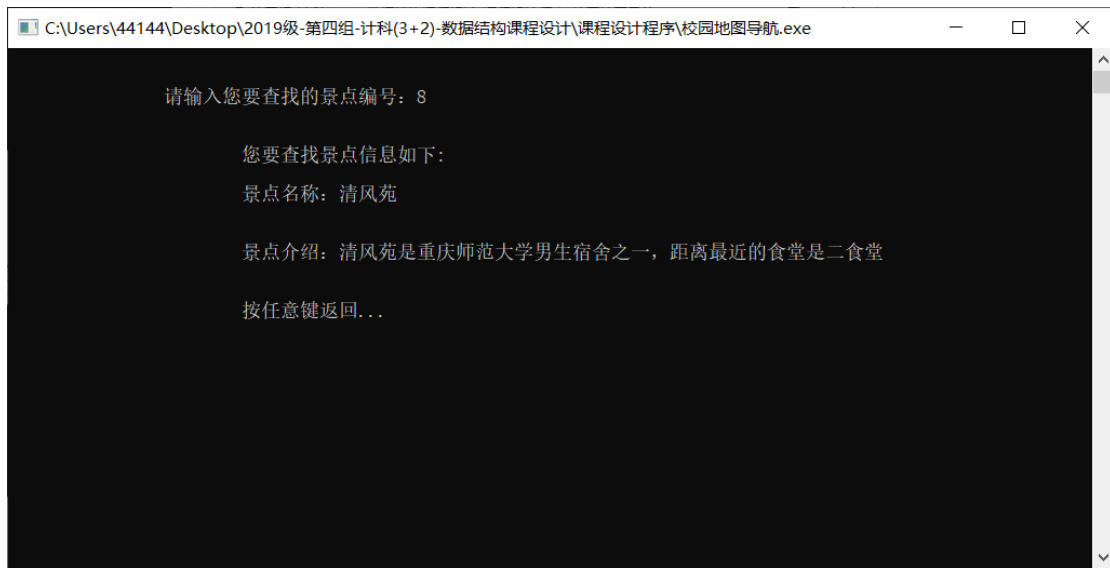
7、选择主菜单中的“6”是退出信息删除，可以根据景点编号或者景点名称进行查询删除：



根据景点编号删除景点：



查询景点 8 为清风苑：



查询景点名字为美术学院没有找到：



根据景点名称删除景点：



查询景点没有找到:



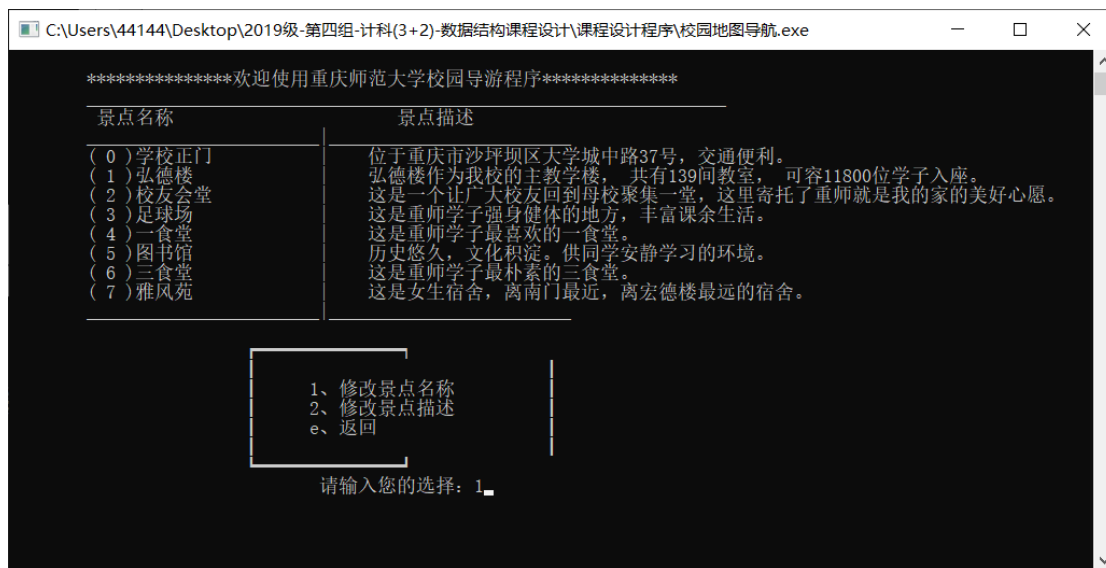
8、选择主菜单中的“7”是景点路径修改，可以修改景点信息，也可以修改道路信息，然后就是修改景点名称或者景点描述：



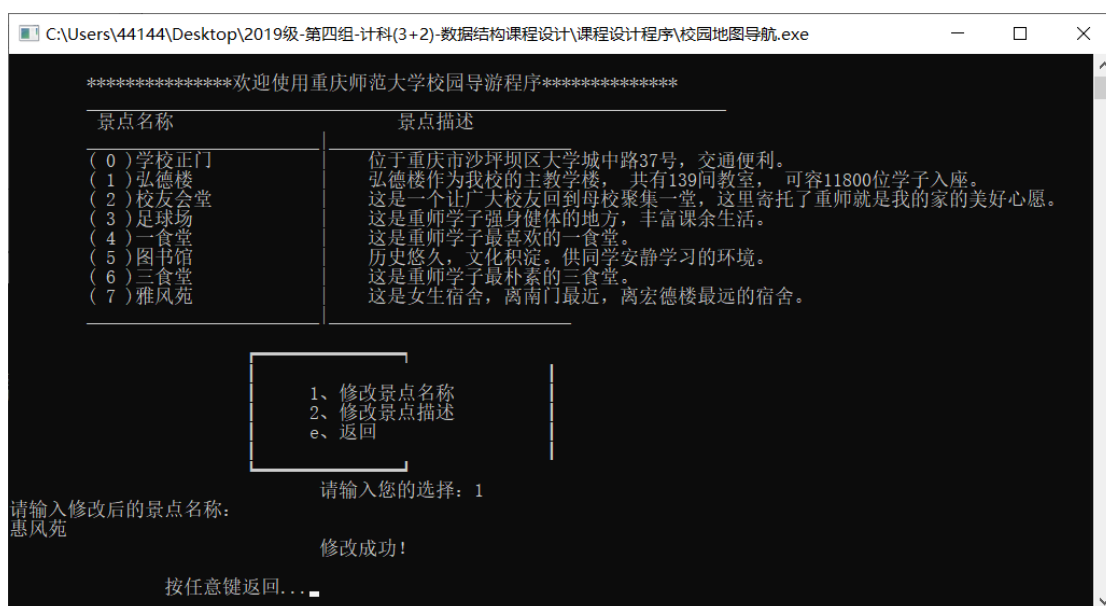
选择需要修改的景点编号：



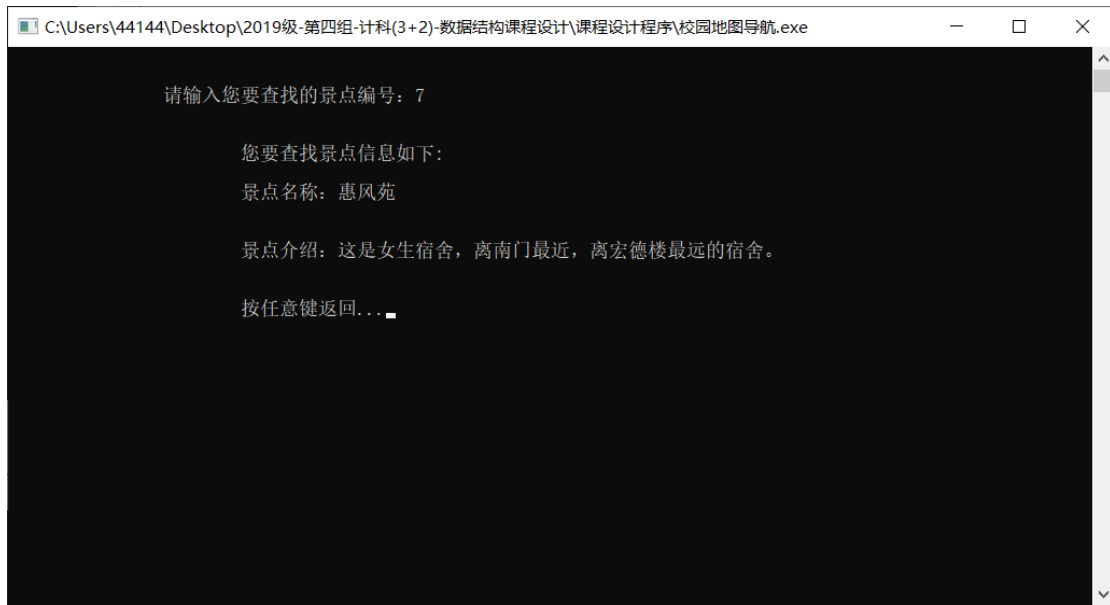
修改景点的名称：



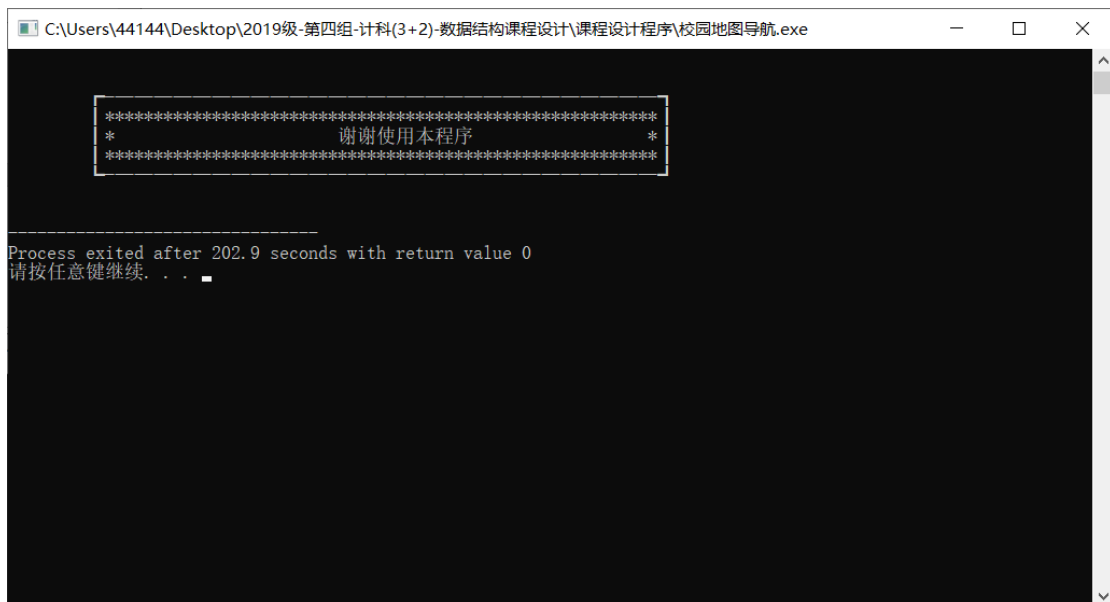
输入修改后的景点名称:



查询修改后的景点信息：



9、选择主菜单中的“8”是退出信息系统。



七、总结

通过本次课程设计，我对图的概念有了一个新的认识，在学习了《数据结构》这门课程之后，我慢慢地体会到了其中的奥妙，图能够在计算机中存在，首先要捕捉他有哪些具体化、数字化的信息，比如说权值、顶点个数等，这也就说明了想要把生活中的信息转化到计算机中必须用数字来完整的构成一个信息库，而图的存在，又涉及到了顶点之间的联系。

经过一个学期对数据结构课程的学习，我能够掌握数据结构所教会我的对待问题的方法，以及遇到问题时如何抽象出一个合理的数据结构类型。数据结构教会我的不但是每一个算法，更多的是如何解决问题的方法。例如，在本次课程设计中我做的是校园导航系统，对于校园导航问题的关键是最短路径的问题，在教材中有算法——迪杰斯特拉求最短路径问题，在花了几天时间后，终于能够将算法的整个流程弄清楚，在对各个定点的存储上采用邻接矩阵的方法，在寻找各个点到其他点的关系的时候更为方便直观。在课程设计中遇到的一系列问题都能够在老师和同学的指导下及时解决。