Project 4
校园导航系统

班级：教务二班

宋渝杰 18340146

廖家源 18340105

刘依澜 18340121

缪贝琪 18340131

【题目要求】

1. 从中山大学东校区的平面图中选取有代表性的景点（15-20 个），抽象成一个无向带权图。以图中顶点表示校内各景点，存放景点名称、代号、简介等信息；以边表示路径，存放路径长度等信息。
2. 为来访客人提供图中任意景点相关信息的查询。
3. 为来访客人提供图中任意景点的问路查询，即查询任意两个景点之间的一条最短的简单路径（区分汽车线路和步行线路）。
4. （额外实现）通过文件处理方式记录景点的查询次数，形成景点的热点值，实现中大景点热门排行。

【数据结构与算法】

数据结构：

构造图的点来表示景点，存放景点名称、简介和热度信息。

```
struct Node{
    string Name;//名字
    string intruction;//简介
    int visitcount;//热度
    Node(){
        Name="";
        intruction="";
        visitcount=0;
    }
    Node(string a,string b,int cou=0){
        Name = a;
        intruction = b;
        visitcount = cou;
    }
};
```

图 1-1 顶点存放信息

分别构建步行和开车的邻接矩阵来表示图，其中若从 a 点到 a 点，记录为 0；从 a 点到 b 点，但二者之间没有直接路径，则记录为 INF；若 ab 间有直接路径，则记录为二者间的距离。

```
int m_dw[MAX][MAX]={ 0  ,180,180,350,INF,INF,INF,INF,INF,INF,INF,INF,INF,INF,INF,INF,//0
                     180,0  ,INF,300,INF,INF,INF,INF,INF,INF,INF,INF,INF,INF,INF,INF,//1
                     180,INF,0  ,200,450,INF,INF,INF,INF,INF,INF,INF,INF,INF,INF,INF,//2
                     350,300,200,0  ,INF,220,120,INF,INF,INF,INF,INF,INF,INF,INF,INF,//3
                     INF,INF,450,INF,0  ,140,INF,140,INF,INF,INF,INF,INF,INF,INF,INF,//4
                     INF,INF,INF,220,140,0  ,220,220,INF,260,INF,INF,INF,INF,INF,INF,//5
                     INF,INF,INF,120,INF,220,0  ,INF,50 ,INF,INF,INF,INF,INF,INF,INF,//6
                     INF,INF,INF,INF,140,220,INF,0  ,INF,260,INF,INF,INF,INF,INF,INF,//7
                     INF,INF,INF,INF,INF,INF,50 ,INF,0  ,370,INF,250,INF,INF,INF,INF,//8
                     INF,INF,INF,INF,INF,260,INF,260,370,0  ,60 ,INF,120,INF,INF,INF,//9
                     INF,INF,INF,INF,INF,INF,INF,INF,INF,60 ,0  ,100,160,INF,INF,INF,//10
                     INF,INF,INF,INF,INF,INF,INF,INF,250,INF,100,0  ,INF,120,INF,INF,//11
                     INF,INF,INF,INF,INF,INF,INF,INF,INF,120,160,INF,0  ,120,200,INF,//12
                     INF,INF,INF,INF,INF,INF,INF,INF,INF,INF,INF,120,120,0  ,120,120,//13
                     INF,INF,INF,INF,INF,INF,INF,INF,INF,INF,INF,INF,200,120,0  ,150,//14
                     INF,INF,INF,INF,INF,INF,INF,INF,INF,INF,INF,INF,INF,120,150,0   //15
```

图 1-2　步行的图的邻接矩阵

```
int m_dc[MAX][MAX]={0  ,180,180,INF,INF,INF,INF,INF,INF,INF,INF,INF,INF,INF,INF,INF,
                    180,0  ,INF,INF,INF,INF,INF,INF,INF,INF,INF,INF,INF,INF,INF,INF,
                    180,INF,0  ,INF,450,INF,INF,INF,INF,INF,INF,INF,INF,INF,INF,INF,
                    INF,INF,INF,0  ,INF,220,INF,INF,INF,INF,INF,INF,INF,INF,INF,INF,
                    INF,INF,450,INF,0  ,140,INF,140,INF,INF,INF,INF,INF,INF,INF,INF,
                    INF,INF,INF,220,140,0  ,220,220,INF,260,INF,INF,INF,INF,INF,INF,
                    INF,INF,INF,INF,INF,220,0  ,INF,INF,INF,INF,INF,INF,INF,INF,INF,
                    INF,INF,INF,INF,140,220,INF,0  ,INF,INF,INF,INF,INF,INF,INF,INF,
                    INF,INF,INF,INF,INF,INF,INF,INF,0  ,370,INF,INF,INF,INF,INF,INF,
                    INF,INF,INF,INF,INF,260,INF,INF,370,0  ,INF,INF,120,INF,INF,INF,
                    INF,INF,INF,INF,INF,INF,INF,INF,INF,INF,0  ,INF,160,INF,INF,INF,
                    INF,INF,INF,INF,INF,INF,INF,INF,INF,INF,INF,0  ,INF,120,INF,INF,
                    INF,INF,INF,INF,INF,INF,INF,INF,120,160,INF,0  ,120,200,INF,
                    INF,INF,INF,INF,INF,INF,INF,INF,INF,INF,INF,120,120,0  ,120,120,
                    INF,INF,INF,INF,INF,INF,INF,INF,INF,INF,INF,INF,200,120,0  ,150,
                    INF,INF,INF,INF,INF,INF,INF,INF,INF,INF,INF,INF,INF,120,150,0
                    };
```

图 1-3　开车的图的邻接矩阵

创建一个主窗口，使得校园导航更加贴近游客。它将提供以下服务选项：

（1）查看中山大学主要景点及其信息；

（2）查询两个地点间的最短路径；

（3）中大景点热门排行；

（4）查看中大模拟地图；

（5）进入到另一个窗口（c#程序）查看更加清晰的导航。

```
void Menu(){
    cout << endl << endl ;
    cout << "\t\t----------------------" << endl;
    cout << "\t\t  *中山大学导游咨询*  " << endl;
    cout << "\t\t----------------------" << endl;
    cout << "\t  1.查看中山大学主要景点及其信息" << endl;
    cout << "\t  2.查询两个地点间的最短路径" << endl;
    cout << "\t  3.中大景点热门排行" << endl;
    cout << "\t  4.查看中大模拟地图" << endl;
    cout << "\t  5.嫌这个太丑？点这里！" << endl;
    cout << "\t  q.退出" << endl;
    cout << "\t----------------------------------" <<endl<<endl;
    cout << "\t请输入对应编码选择您所需的操作：  " ;
}
```

图 1-4　主窗口内容

算法：

1. 查看中山大学主要景点及详细信息的 Introduce 函数：首先，函数输出各个景点的编号和名称；用户可以通过输入对应的编号来查看想要了解的地点的详细信息（其中输入 0 退出查询回到主菜单）。

2. 输出最短路径及长度的 GetRoute 函数：分别对步行及开车的邻接矩阵做下列操作，不妨将邻接矩阵先记作 m。用 int 类型 x1,x2 分别别代表起点和终点在数组 name 中对应的下标；创建 int 类型的 Dis 数组来记录该起点到其他

各个点的最短距离的长度；创建 string 类型的 DD 数组来存储起点到其他各点的最短路径（e.g. a->b->c）；创建 bool 类型的 S1 数组，某位为 1 表示该位已经确定了最短距离及路径，为 0 则还没有。Dis 初始化为起点在邻接矩阵中对应的那一行；DD 初始化为空；S1 初始化为 0。接着将起点放入已确定的部分，即 S1[x1]=1，其路径也确定为其本身，输入 DD[x1]中。创建 int 类型的 which1 和 which2，which2 用来表示遍历后得到的最短路径的对应的终点下标，which1 表示 which2 的最短路径是从下标 which1 对应的最短路径加上 which1 对应的点到终点的边。然后进行循环：定义 int 类型 MIN，初始化成 INF，用来比较已有的直接路径来确定最短的一条。下一个能确定最短路径的点一定是由已经确定最短路径的点延伸一条边出去得到的。我们通过筛选满足条件的点，并不断刷新这些点到起点的距离，以及通过 MIN 的比较来确定最小值以及 which1 和 which2。得到的 which2 就为已经确定最短路径的点，而其最短的路径长度就存在对应的 Dis 中，路径则为 which1 的路径加上->name[which2]->Name。该循环直至当终点也纳入了已确定部分（即 S1[x2]==1 时）停止。最后，输出最短距离及其路径。

3. 文件处理：本程序使用了文件（景点热度.txt）来记录景点的热点信息：当程序打开时，读取文件的热点信息并存入系统中；而当某个景点被查询一次时，热点+1，并保存至文件中。

```
ifstream read("景点热度.txt",ios::in);
ofstream write("景点热度.txt",ios::out);
```

图 1-5 文件处理关键代码

4. 利用 windows.h 库中的函数 system 可以打开中山大学的模拟地图，以及用 C#做的更加美观易操作的窗口。

```
system("模拟地图.png");
system("c#项目4.exe");
```

图 1-6 调用文件/程序关键代码

5. c#部分：在程序通过上述 system 调用“项目 4 小玩具.exe”之后，便会打开另一个程序，该程序用 c#编写，具有景点查询、最短路径功能，同时能显示直观的图，以及各个景点的信息和照片：

图 1-7 c#程序主界面

景点信息：程序左侧的图中的每一个景点均为 button（按钮）控件，点击后程序中间偏上部分“景点信息”组件内将会显示景点的名称和简介，右侧的“景点图片”组件内将会显示景点的照片（由小组成员亲自拍摄）。

最短路径：程序中间偏下部分的“最短路径”组件中，用户通过在地标 1 和地标 2 这两个下拉框中选择两个地点，组件内便会显示开车、步行距离、路程耗时、以及具体的路线。

（开车时速以 20km\h 计，步行时速以 4km\h 计）。

异常处理：

当用户输入错误信息时（例如该输入数字时，用户输入了字符），系统调用 cin.good()函数判断输入错误，提示“输入错误，请重新输入”，然后调用 cin.clear()、cin.sync()函数清除输入流的缓存，并让用户重新输入。

【测试数据、结果及分析】

C++部分：

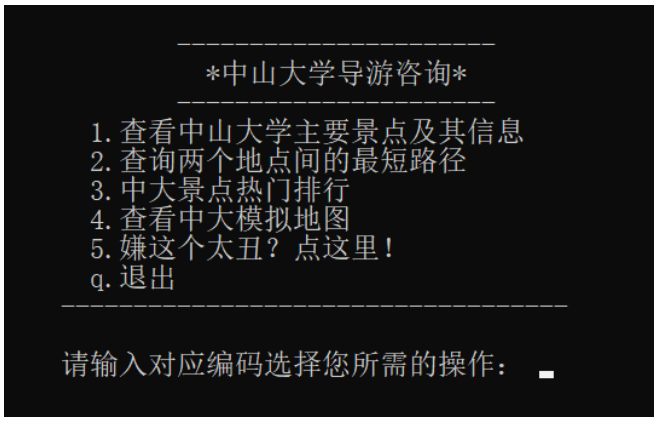

图 2-1 菜单

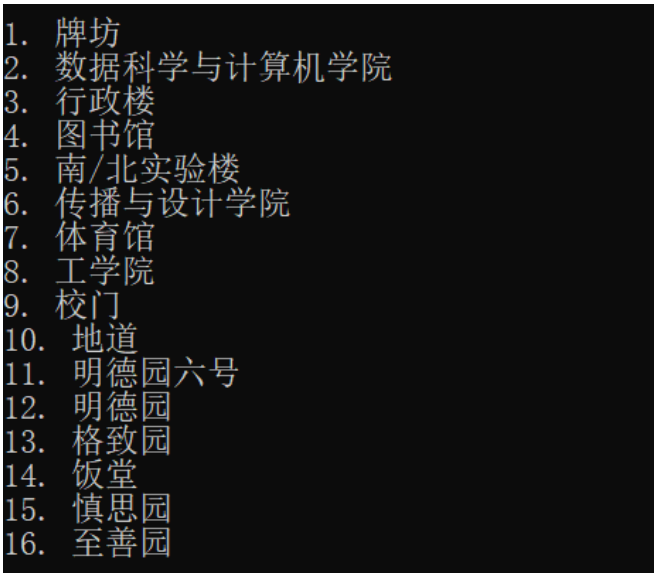

图 2-2 输入 1，查看中山大学的主要景点

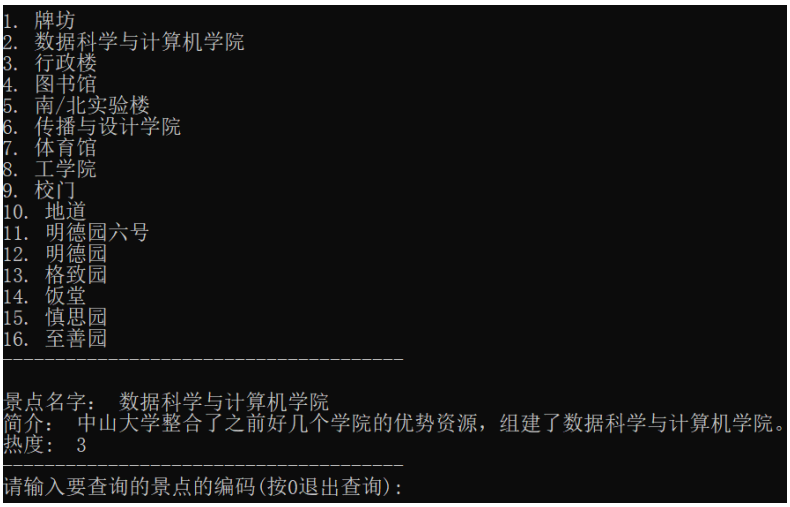

图 2-3 输入要查询景点信息的编码，查看信息

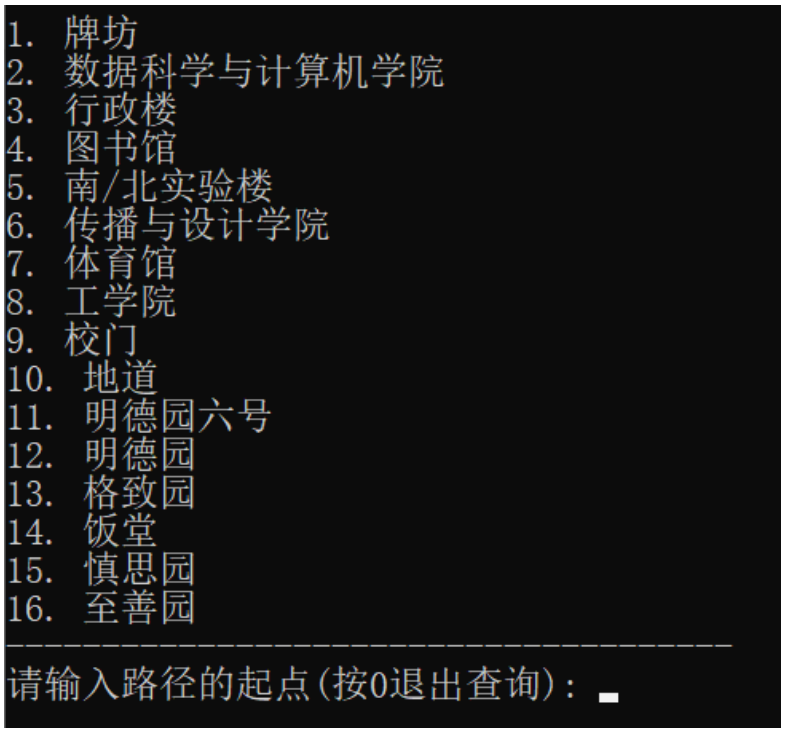

图 2-4 在菜单输入 2，查询两点最短路径

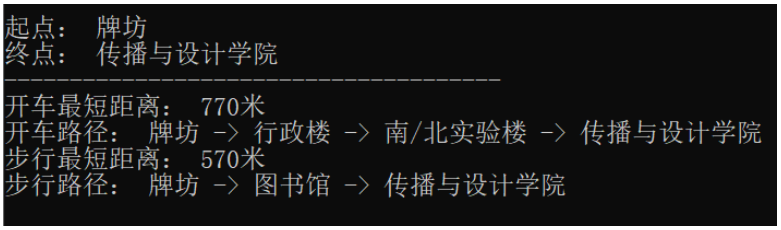

图 2-5 输入起点终点，输出开车与步行的最短距离和路径

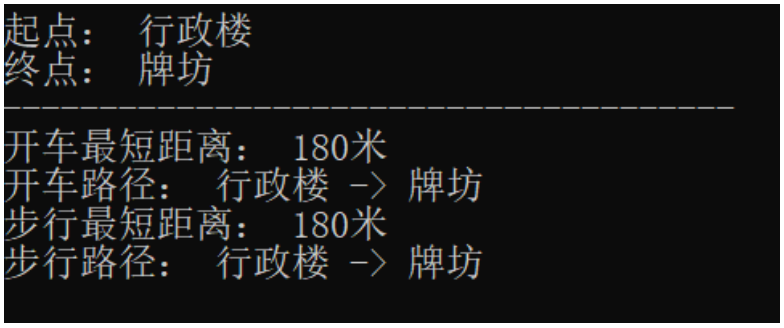

图 2-6 再次输入起点终点，查看距离与路径

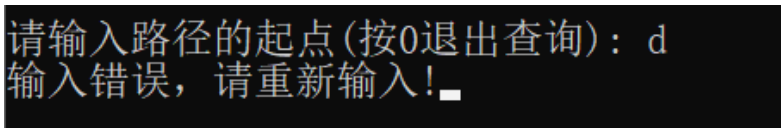

图 2-7 异常处理，输入错误，重新输入

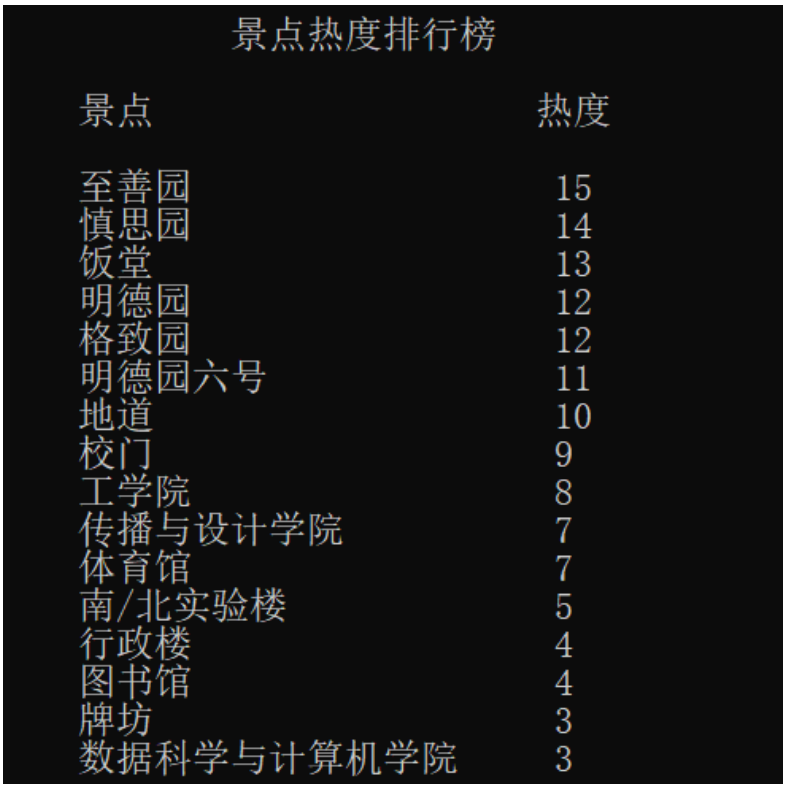

图 2-8 在菜单输入 3，查看景点热度排行榜

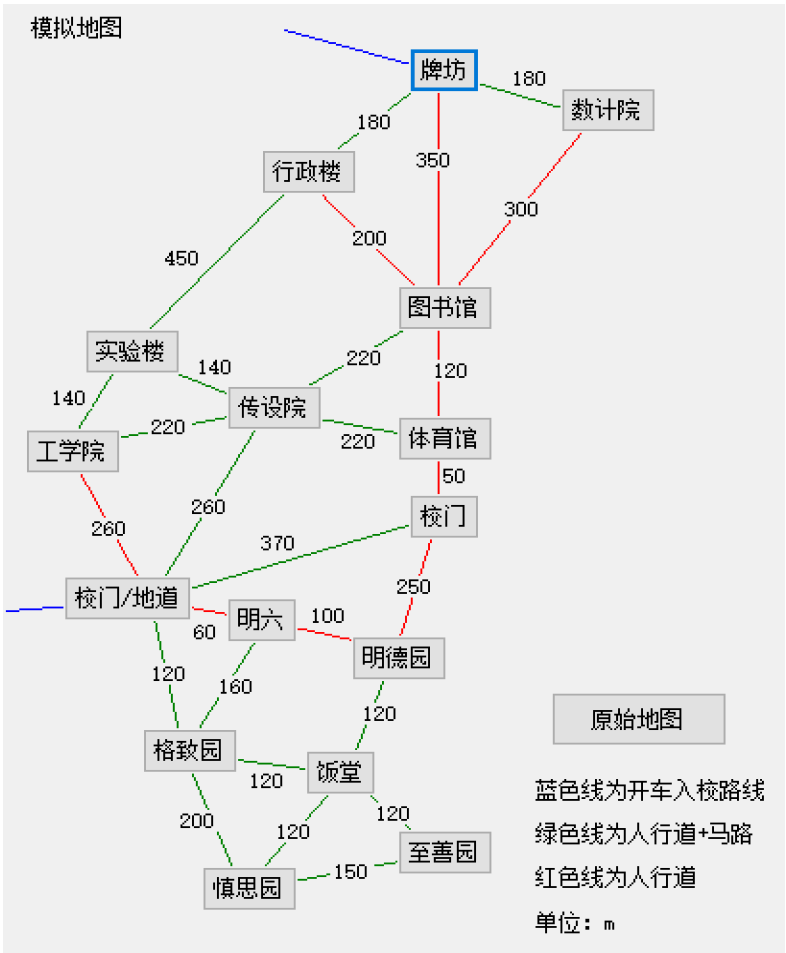

图 2-9 在菜单输入 4，弹出弹窗，显示中大模拟地图

图 2-10 在菜单输入 5，系统调用 c#程序

C#部分：

1. 在左侧地图中点击“数计院”按钮，“景点信息”组件中显示对应的名称、简介，“景点图片”组件中显示对应的照片。

图 3-1 点击“数计院”按钮

2. 在“最短路径”组件中，在地标 1、地标 2 的下拉框中选择相应地点，选择完毕后程序输出开车、步行距离，大致路程时间，以及具体的路线：

图 3-2 “地标 2”下拉框选择过程

图 3-3 选择完毕后显示最短路径

【分工、贡献%、自我评分】

分工：

宋渝杰： C#程序的设计和实现，C++代码的后期修改，25%，99 分

廖家源： C++菜单、文件操作、异常处理的代码编写，25%，99 分

刘依澜： C++最短路径函数的编写，实验报告的撰写，25%，99 分

缪贝琪： C++查询景点函数的编写，实验报告的撰写，25%，99 分

【项目总结】

在这次的项目四中，虽然表面上说是一个小项目，但是在实现的方面也有许多重点和难点，比如说直观图的显示：在 c++黑框程序中显示直观图不仅难度较高，而且也十分不美观。考虑到这个图在程序中是不会被修改的，我们采用了 system(“图片文件名.png”)的方式，事先做好一份美观的校园图的 png 格式图片，再在 c++程序中使用上述代码直接打开 png 格式图片来显示直观图。

项目四相比于项目三来说，我们小组在技术上也有了不错的提升，比如说在这次项目中，采用了 system(“c#程序.exe”)的代码，完成了 c++与 c#程序之间的接口，使得可以直接在 c++程序中打开 c#程序。而在项目三中，我们的 c++和 c#程序还是完全独立的两部分，这是我们在项目四中的一个技术上的提升。

对于小组成员分工合作方面，我们在周五项目出来当天就规划好了分工内容，先由我完成了相对直观的 c#程序，再由两位女成员按照 c#程序的内容完成了 c++程序的初版，满足了景点查询和最短路径的基本要求。在时间还相对充裕的情况下，我们加入了“景点的热点值”,采用了文件处理技术。最后一起上网学习了 system()函数，实现直观图的图片形式输出和两个程序的接口，然后形成了现在的终版。

总的来说，这次项目四我们学习了许多程序代码方面的技术，时间意识、小组合作意识也比项目二有了质的提升，希望以后的合作中再接再厉。

【程序清单】

c++程序：（c#程序在代码压缩包中附带源码，此处只列出 c++程序源码）

```
#include <iostream>
#include <vector>
#include <stack>
#include <conio.h>
#include <fstream>
#include <iomanip>
#include <time.h>
#include <windows.h>
const int MAX = 16;
const int INF = 10000 ;
using namespace std;

struct Node{
    string Name;
```

```
    string intruction;
    int visitcount;
    Node(){
        Name="";
        intruction="";
        visitcount=0;
    }
    Node(string a,string b,int cou=0){
        Name = a;
        intruction = b;
        visitcount = cou;
    }
};

Node* name[MAX];
int
m_dw[MAX][MAX]={ 0   ,180,180,350,INF,INF,INF,INF,INF,INF,INF,INF,INF,I
NF,INF,INF,//0

    180,0    ,INF,300,INF,INF,INF,INF,INF,INF,INF,INF,INF,INF,INF,//
1

    180,INF,0    ,200,450,INF,INF,INF,INF,INF,INF,INF,INF,INF,INF,//
2

    350,300,200,0    ,INF,220,120,INF,INF,INF,INF,INF,INF,INF,INF,//
3

    INF,INF,450,INF,0    ,140,INF,140,INF,INF,INF,INF,INF,INF,INF,//
4

    INF,INF,INF,220,140,0    ,220,220,INF,260,INF,INF,INF,INF,INF,//
5

    INF,INF,INF,120,INF,220,0   ,INF,50 ,INF,INF,INF,INF,INF,INF,//6
```

```
    INF,INF,INF,INF,140,220,INF,0    ,INF,260,INF,INF,INF,INF,INF,INF,//
7

    INF,INF,INF,INF,INF,INF,50 ,INF,0   ,370,INF,250,INF,INF,INF,INF,//8

    INF,INF,INF,INF,INF,260,INF,260,370,0   ,60 ,INF,120,INF,INF,INF,//9

    INF,INF,INF,INF,INF,INF,INF,INF,INF,60  ,0    ,100,160,INF,INF,INF,//
10

    INF,INF,INF,INF,INF,INF,INF,INF,250,INF,100,0    ,INF,120,INF,INF,//
11

    INF,INF,INF,INF,INF,INF,INF,INF,INF,120,160,INF,0    ,120,200,INF,//
12

    INF,INF,INF,INF,INF,INF,INF,INF,INF,INF,INF,120,120,0    ,120,120,//
13

    INF,INF,INF,INF,INF,INF,INF,INF,INF,INF,INF,INF,200,120,0    ,150,//
14

    INF,INF,INF,INF,INF,INF,INF,INF,INF,INF,INF,INF,INF,120,150,0
//15
                        };
int
m_dc[MAX][MAX]={0   ,180,180,INF,INF,INF,INF,INF,INF,INF,INF,INF,INF,INF,IN
F,INF,INF,

    180,0   ,INF,INF,INF,INF,INF,INF,INF,INF,INF,INF,INF,INF,INF,INF,

    180,INF,0   ,INF,450,INF,INF,INF,INF,INF,INF,INF,INF,INF,INF,INF,

    INF,INF,INF,0   ,INF,220,INF,INF,INF,INF,INF,INF,INF,INF,INF,INF,

    INF,INF,450,INF,0   ,140,INF,140,INF,INF,INF,INF,INF,INF,INF,INF,
```

```
    INF,INF,INF,220,140,0  ,220,220,INF,260,INF,INF,INF,INF,INF,INF,

    INF,INF,INF,INF,INF,220,0  ,INF,INF,INF,INF,INF,INF,INF,INF,INF,

    INF,INF,INF,INF,140,220,INF,0  ,INF,INF,INF,INF,INF,INF,INF,INF,

    INF,INF,INF,INF,INF,INF,INF,INF,0  ,370,INF,INF,INF,INF,INF,INF,

    INF,INF,INF,INF,INF,260,INF,INF,370,0  ,INF,INF,120,INF,INF,INF,

    INF,INF,INF,INF,INF,INF,INF,INF,INF,INF,0  ,INF,160,INF,INF,INF,

    INF,INF,INF,INF,INF,INF,INF,INF,INF,INF,INF,0  ,INF,120,INF,INF,

    INF,INF,INF,INF,INF,INF,INF,INF,INF,120,160,INF,0  ,120,200,INF,

    INF,INF,INF,INF,INF,INF,INF,INF,INF,INF,INF,120,120,0  ,120,120,

    INF,INF,INF,INF,INF,INF,INF,INF,INF,INF,INF,INF,200,120,0  ,150,

    INF,INF,INF,INF,INF,INF,INF,INF,INF,INF,INF,INF,INF,120,150,0
                    };

bool Find(int a,int s[100],int d); //判断a在不在s里
void Menu();      // 输出主菜单
void Initial();   // 程序启动的初始化操作

void Introduce();//地点简介
void GetRoute(); //查询最短路径
void Rank();      //景点访问排行榜
void FreeSpace();    //结束程序时回收内存空间
void FileRead(); // 读取文件信息
void FileWrite();// 修改并关闭文件

int main(){
```

```
    Initial();
    while(1){
        Menu();
        char x;
        do{
            x=_getch();
        } while((x<'1' or x>'5') and x!='q' and x!='Q');
        if(x=='q' or x=='Q'){
            FileWrite();
            FreeSpace();
            break;
        }
        system("cls");
        switch(x){
            case '1':
                Introduce();
                break;
            case '2':
                GetRoute();
                break;
            case '3':
                Rank();
                break;
            case '4':
                system("模拟地图.png");
                break;
            case '5':
                system("c#项目 4.exe");
                break;
        }
        system("cls");
    }
}
```

bool Find(int a,int s[100],int d) //a：看 a 在不在 s 里，d：s 共有多少个元素

```
{
    for(int i = 0; i < d; i++)
    {
        if(s[i] == a)
            return true;
    }
    return false;
}

void Menu(){
    cout << endl << endl ;
    cout << "\t\t----------------------" << endl;
    cout << "\t\t  *中山大学导游咨询*  " << endl;
    cout << "\t\t----------------------" << endl;
    cout << "\t  1.查看中山大学主要景点及其信息" << endl;
    cout << "\t  2.查询两个地点间的最短路径" << endl;
    cout << "\t  3.中大景点热门排行" << endl;
    cout << "\t  4.查看中大模拟地图" << endl;
    cout << "\t  5.嫌这个太丑？点这里！" << endl;
    cout << "\t  q.退出" << endl;
    cout << "\t-----------------------------------" <<endl<<endl;
    cout << "\t 请输入对应编码选择您所需的操作：  " ;
}
void Initial(){
    name[0]=new Node("牌坊","整座门楼用钢筋三合土筑成，外饰面用花岗石砌成。高 10.98 米，宽 25.315 米。门额刻有“国立中山大学”六个大字");
    name[1]=new Node("数据科学与计算机学院","中山大学整合了之前好几个学院的优势资源，组建了数据科学与计算机学院。");
    name[2]=new Node("行政楼","行政办公的地方");
    name[3]=new Node("图书馆","又称中东自习室，位置不多，晚点来不好找位置做作业；曾经是中东目测最大建筑，不过现在被啥化学学院楼取代了");
    name[4]=new Node("南/北实验楼","同学们平时打码、做祭祖等其他实验的地方，以及许多老师的办公室和实验室也设在里面");
    name[5]=new Node("传播与设计学院","是聚集了一群优质妹子和强势女篮的天堂，但由于院楼构造奇特，又被称为“中东马桶楼”");
    name[6]=new Node("体育馆","除了第一节体育课用来集合之外，还可以用作
```

```
礼堂和其他活动场所，租金好像是一天两万");
  name[7]=new Node("工学院","除了建筑面前的小广场可以用来举办活动之外，还真不知道有啥其他用处");
   name[8]=new Node("校门","学生们上学的唯二通道。曾经因为中大修校门事件被称为“狗洞”，但是其实说的也挺合理的");
   name[9]=new Node("地道","学生们上学的唯二通道，单车党飙车圣地，但是爬坡的时候就会特别累");
   name[10]=new Node("明德园六号","聚集了快递点、理发店、银行等众多日常设施的地方，偶尔也能申请三楼房间用于搞活动");
   name[11]=new Node("明德园","中东宿舍楼之一,特点是离教学区和明六比较近");
   name[12]=new Node("格致园","中东宿舍楼之一,特点是供奉着深圳校区的学生们");
   name[13]=new Node("饭堂","大致分为四个饭堂，一、二、四饭比较普通，三饭比较好吃比较贵(并不是");
   name[14]=new Node("慎思园","中东宿舍楼之一,特点是南校门封了之后就变得比较偏僻，注意慎思园是没有 1-4 号的哦");
   name[15]=new  Node("至善园","中东宿舍楼之一，特点是住着我们院的女生!!!! ");
   FileRead();
}

void Introduce(){
   int a;
   bool flag = false;  //控制界面显示
   Int:
   system("cls");
   cout<<"\n";
   for(int i=0;i<MAX;i++){
      cout << i+1 << ". " << name[i]->Name << endl;
   }
   cout << "--------------------------------------" << endl;
   if(flag){
      name[a-1]->visitcount+=1;
      cout << endl << "景点名字:  " << name[a-1]->Name << endl;
      cout << "简介:  " << name[a-1]->intruction << endl;
```

```
        cout << "热度:  " << name[a-1]->visitcount <<endl;
        cout << "-------------------------------------" << endl;
    }
    cout << "请输入要查询的景点的编码(按 0 退出查询): ";
    cin >> a;
    if(!cin.good() or a<0 or a>16){
        cout << "输入错误，请重新输入!";
        cin.clear();
        cin.sync();
        Sleep(500);
        flag = false;
        goto Int;
    }
    if(a==0) return;
    else{
        flag = true;
        cin.clear();
        cin.sync();
        goto Int;
    }

}
void GetRoute(){
    int sta; //起点
    int dst; //终点
    bool flag1 = false;  //控制界面显示
    bool flag2 = false;  //控制界面显示
    GRT:
    system("cls");
    cout<<"\n";
    for(int i=0;i<MAX;i++){
        cout << i+1 << ". " << name[i]->Name << endl;
    }
    cout << "-------------------------------------" << endl;
    if(!flag1){
        cout << "请输入路径的起点(按 0 退出查询): ";
```

```
    cin >> sta;
    if(!cin.good() or sta<0 or sta>16){
        cout << "输入错误，请重新输入!";
        cin.clear();
        cin.sync();
        Sleep(500);
        flag1 = false;
        goto GRT;
    }
    if(sta==0) return;
    else{
        flag1 = true;
        cin.clear();
        cin.sync();
        goto GRT;
    }
}else if(flag1 and !flag2){
    cout << "起点：  "<<name[sta-1]->Name <<endl;
    cout << "-------------------------------------" << endl;
    cout << "请输入路径的终点(按 0 退出查询): ";
    cin >> dst;
    if(!cin.good() or dst<0 or dst>16){
        cout << "输入错误，请重新输入!";
        cin.clear();
        cin.sync();
        Sleep(500);
        flag2 = false;
        goto GRT;
    }
    if(dst==0) return;
    else{
        flag2 = true;
        cin.clear();
        cin.sync();
        goto GRT;
    }
```

```
    }else if(flag1 and flag2){
        cout << "起点:  "<<name[sta-1]->Name <<endl;
        cout << "终点:  "<<name[dst-1]->Name <<endl;
        cout << "--------------------------------------" << endl;

        //搞个车车!!!
        int x1 = sta-1;
        int x2 = dst-1;
        int Dis[MAX];//距离, 会不断刷新
        bool S1[MAX]={0};
        string DD[MAX];//show the path
        for(int i=0;i<MAX;i++){
            Dis[i]=m_dc[x1][i];
            S1[i]=false;
            DD[i]="";
        }
        S1[x1]=1;
        DD[x1]=name[x1]->Name;
        int which1=0;
        int which2=0;
        while(!S1[x2]){
            int MIN=INF;
            int i,j;
            for(i=0;i<MAX;i++){
                for(j=0;j<MAX;j++){
                    if(S1[i]==1  &&  m_dc[i][j]!=INF  &&  m_dc[i][j]!=0  &&
S1[j]==0){
                        if(Dis[j]>Dis[i]+m_dc[i][j])
                            Dis[j]=Dis[i]+m_dc[i][j];
                        if(MIN>Dis[j]){
                            MIN=Dis[j];
                            which1=i;
                            which2=j;
                        }
                    }
                }
```

```
            }
            S1[which2]=1;
            DD[which2]=DD[which1]+ " -> " + name[which2]->Name;
        }
        cout << "开车最短距离:  " << Dis[x2] << "米" << endl;
        cout << "开车路径:  " << DD[x2] << endl;



        for(int i=0;i<MAX;i++){
            Dis[i]=m_dw[x1][i];
            S1[i]=false;
            DD[i]="";
        }
        S1[x1]=1;
        DD[x1]=name[x1]->Name;
        which1=0;
        which2=0;
        while(!S1[x2]){
            int MIN=INF;
            int i,j;
            for(i=0;i<MAX;i++){
                for(j=0;j<MAX;j++){
                    if(S1[i]==1  &&  m_dw[i][j]!=INF  &&  m_dw[i][j]!=0  &&
S1[j]==0){
                        if(Dis[j]>Dis[i]+m_dw[i][j])
                            Dis[j]=Dis[i]+m_dw[i][j];
                        if(MIN>Dis[j]){
                            MIN=Dis[j];
                            which1=i;
                            which2=j;
                        }
                    }
                }
            }
            S1[which2]=1;
```

```
            DD[which2]=DD[which1]+ " -> " + name[which2]->Name;
        }

        cout << "步行最短距离： " << Dis[x2] << "米" << endl;
        cout << "步行路径： " << DD[x2] << endl;

        cout<<"\n\n\t\t\t按任意键继续...";
        _getch();
        flag1 = false;
        flag2 = false;
        name[sta-1]->visitcount+=1;
        name[dst-1]->visitcount+=1;
        goto GRT;
    }
}

void Rank(){
    Node* rank[MAX];
    for(int i=0;i<MAX;i++)
        rank[i] = name[i];
    // 从大到小排序
    for(int i=0;i<MAX-1;i++){
        for(int j=0;j<MAX-i-1;j++){
            if(rank[j]->visitcount<rank[j+1]->visitcount){
                Node* tmp = rank[j];
                rank[j] = rank[j+1];
                rank[j+1] = tmp;
            }
        }
    }
    cout<<"\n\n";
    cout<<"\t\t景点热度排行榜\n\n";
    cout<<"\t景点\t\t\t热度\n\n";
    for(int i=0;i<MAX;i++){
        cout.setf(ios::left);
        cout<<"\t"<<setw(25)<<rank[i]->Name<<rank[i]->visitcount<<endl;
```

```
    }
    cout<<"\n\n\t\t\t\t 按任意键放回...";
    _getch();
}

void FileRead(){
    ifstream read("景点热度.txt",ios::in);
    if(!read){
        ofstream write("景点热度.txt",ios::out);
        if(write.is_open()){
            for(int i=0;i<MAX;i++){
                write<<name[i]->Name<<"\t"<<name[i]->visitcount<<endl;
            }
            write.close();
        }
    }else{
        int cou;
        string na;
        for(int i=0;i<MAX;i++){
            read>>na>>cou;
            name[i]->visitcount = cou;
        }
        read.close();
    }
}
void FileWrite(){
    ofstream write("景点热度.txt",ios::out);
    if(write.is_open()){
        for(int i=0;i<MAX;i++){
            write<<name[i]->Name<<"\t"<<name[i]->visitcount<<endl;
        }
        write.close();
    }
}
void FreeSpace(){
    for(int i=0;i<MAX;i++)
```

```
        delete name[i];
}
```