#include<stdio.h>
  
#include <stdlib.h>
  
#include<string.h>
  
#define INFINITY 10000
  
#define M 40 /\*最大顶点数\*/
  
typedef struct {
  
 int num; /\*景点代号\*/
  
 char name[20]; /\*景点名称\*/
  
 char intro[200]; /\*景点简介\*/
  
}information;
  
   
typedef int edgtype; /\*权值类型\*/
  
typedef struct { /\*校园景点图结构体\*/
  
 information vexs[M]; /\*顶点信息域\*/
  
 edgtype edge[M][M]; /\*邻接矩阵\*/
  
 int vexNum, edgNum; /\*顶点数和边数\*/
  
}matrix;
  
   
int menu(); /\*主菜单\*/
  
void Createmap(matrix \*g); /\*从文件读取信息建立图\*/
  
void Print\_Map(); /\*显示校园景点地图\*/
  
int Judge(int num); /\*判断输入的编号是否合理\*/
  
void searchinfo(matrix \*g); /\*景点信息查询\*/
  
void shortPath(matrix \*g); /\*求景点间最短路径\*/
  
void Floydprint(matrix \*g, int s, int e); /\*递归打印两点间最短路径\*/
  
void shortpath\_Print(matrix \*g); /\*输出并打印两点间的最短路径\*/
  
void Dfs(matrix \*g, int s, int e); /\*深度优先遍历查询两景点间所有路径\*/
  
void Allpath\_Print(matrix \*g); /\*查询两顶点间的所有路径并打印\*/
  
void BestPath(matrix \*g); /\*多顶点间求最佳路径\*/
  
void Exit(); /\*退出系统\*/
  
void Information\_print(); /\*公告栏的打印\*/
  
void Manager(); /\*管理员界面\*/
  
int Login\_system(); /\*系统主界面\*/
  
void scenicManager(matrix \*g); /\*对景点的管理\*/
  
int Login(); /\*管理员登录\*/
  
void searchinfo(matrix \*g); /\*景点信息查询\*/
  
   
/\*校园景点图的读取与创建\*/
  
void Createmap(matrix \*g)
  
{
  
 int i, j, k, e;
  
 FILE \*rf;
  
 rf = fopen("map.txt", "r");
  
 if (rf)
  
 {
  
 fscanf(rf, "%d%d", &g->vexNum, &g->edgNum);
  
 for (i = 0; i < g->vexNum; i++)
  
 fscanf(rf, "%d%s%s", &g->vexs[i].num, g->vexs[i].name, g->vexs[i].intro);
  
 for (i = 0; i < g->vexNum; i++)
  
 for (j = 0; j < g->vexNum; j++)
  
 {
  
 if (i == j)
  
 g->edge[i][j] = 0;
  
 else
  
 g->edge[i][j] = INFINITY;
  
 }
  
 for (k = 0; k < g->edgNum; k++)
  
 {
  
 fscanf(rf, "%d%d%d", &i, &j, &e);
  
 g->edge[i-1][j-1] = g->edge[j-1][i-1] = e;
  
 }
  
 fclose(rf);
  
 }
  
 else
  
 g->edgNum = 0;
  
}
  
   
/\*校园景点图的显示\*/
  
void Dis\_Map() {
  
 printf("\n 『广东海洋大学校园景点地图一览』 \n\n");
  
 printf(" \*200\* \*200\* \*150\* \*100\* \*100\* \n");
  
 printf("(34)西区宿舍=============(33)第四饭堂====================(32)西区体育场====(29)校医院=============(30)林果楼===========(31)工程训练中心==================== \n");
  
 printf(" || \*200\*|| ||\*250\* \n");
  
 printf(" || || \*80\* || \n");
  
 printf(" || (27)海科楼====(28)第三实验楼 || \n");
  
 printf(" || \*300\* || || \n");
  
 printf(" || \*50\* || ============== \n");
  
 printf(" || \*150\* (26)科技楼 || 东 || \n");
  
 printf(" || (25)第三食堂================|| \*80\* || 区 || \n");
  
 printf(" || \*300\* \*100\* || \*150\*(22)钟海楼A \*50\* \*50\* \*30\* \*100\* || ||\*100\* \n");
  
 printf("(35)艺术楼===========================================(24)第二教学楼 ===========||\*10\*=======(16)中心广场========(17)图书馆========(18)体育馆========|| ||=======(20)第二饭堂 \n");
  
 printf(" || || (23)钟海楼B || || || || (19) || || \n");
  
 printf(" || \*100\*|| ||\*30\* || ||\*40\* || || \*150\*|| \n");
  
 printf(" || (13)教工食堂 (14)主教学楼 ||\*50\* || || || || \n");
  
 printf(" || \*300\* || || || || || 宿 || (21)体育场 \n");
  
 printf(" || ||\*100\* \*5\* \*40\*|| \*5\* || \*30\* || \*80\* || 舍 || || \n");
  
 printf(" || \*200 \* (12)教工宿舍=========(7)兴农楼===========(9)海洋广场=======(8)兴海楼===========第一饭堂(15)========================= || \n");
  
 printf("(36)试验田====================================/ || || || 商业中心(15) \*200\*|| \n");
  
 printf(" || \*200 ||\*150\* ||\*20\* \*50\* \*30\*|| \*20\* ||\*30\* || \n");
  
 printf(" ||===================================(11)水生博物馆===========(5)行政楼=============================(6)兴教楼==================(10)水产楼 || \n");
  
 printf(" || || \*200\* \*50\* || \*50\* \*60\* || \*10\* || \*30\* ||\n");
  
 printf(" || || \*250\* ||=======================================================||============================= \n");
  
 printf(" ||=======================================================================(2)蝴蝶湖 || 蝴蝶湖 \*80\* (3)游泳池 \*50\* (4) 网球场 \n");
  
 printf(" \*400\* \*10\* || \n");
  
 printf(" (1) 广东海洋大学正门口 \n\n");
  
   
   
   
}
  
   
/\*景点信息查询\*/
  
void searchinfo(matrix \*g) {
  
 int s;
  
 do {
  
 printf("\n请输入你要查找的景点编号：");
  
 scanf("%d", &s);
  
 } while (Judge(s));
  
 printf("\n景点名称：[%s]\n\n", g->vexs[s - 1].name);
  
 printf("景点简介： %s\n\n", g->vexs[s - 1].intro);
  
}
  
   
/\*Floyd算法求两景点间的一条最短的路径\*/
  
int dist[M][M]; /\*距离向量\*/
  
int path[M][M]; /\*路径向量\*/
  
void shortPath(matrix \*g) {
  
 int i, j, k;
  
 for (i = 0; i < g->vexNum; i++) /\*初始化距离向量矩阵与路径向量矩阵\*/
  
 for (j = 0; j < g->vexNum; j++) {
  
 dist[i][j] = g->edge[i][j];
  
 if (i != j && dist[i][j] < INFINITY) path[i][j] = i;
  
 else path[i][j] = -1; /\*-1代表当前两点不可达\*/
  
 }
  
 for (k = 0; k < g->vexNum; k++) /\*递推求解每两景点的最短路径\*/
  
 for (i = 0; i < g->vexNum; i++)
  
 for (j = 0; j < g->vexNum; j++) /\*更新dist[i][j]的值\*/
  
 if (dist[i][j] >(dist[i][k] + dist[k][j])) {
  
 dist[i][j] = dist[i][k] + dist[k][j];
  
 path[i][j] = k; /\*path用于记录最短路径上的经结点\*/
  
 }
  
}
  
   
/\*递归实现打印两点间的最短路径\*/
  
void Floydprint(matrix \*g, int s, int e) {
  
 if (path[s][e] == -1 || path[s][e] == e || path[s][e] == s)
  
 return;
  
 else {
  
 Floydprint(g, s, path[s][e]); /\*将中间点作为终点继续打印路径\*/
  
 printf("%s->", g->vexs[path[s][e]].name); /\*打印中间景点名字\*/
  
 Floydprint(g, path[s][e], e); /\*将中间点作为起点继续打印路径\*/
  
 }
  
}
  
   
/\*输出并打印两点间的最短路径\*/
  
void shortpath\_Print(matrix \*g) {
  
 int s, e; /\*起点编号，终点编号\*/
  
 do {
  
 printf("\n请输入起点编号：");
  
 scanf("%d", &s);
  
 } while (Judge(s));
  
 do {
  
 printf("\n请输入终点编号：");
  
 scanf("%d", &e);
  
 } while (Judge(e));
  
 printf("\n%s到%s的最短距离是：%dm\n", g->vexs[--s].name, g->vexs[--e].name, dist[s-1][e-1]);
  
 printf("\n这是最佳游览路线：");
  
 printf("%s->", g->vexs[s].name); /\*输出路径上的起点\*/
  
 Floydprint(g, s, e); /\*输出路径上的中间点\*/
  
 printf("%s\n\n", g->vexs[e].name); /\*输出路径上的终点\*/
  
}
  
   
   
/\*深度优先遍历查询任意两个景点之间的所有路径\*/
  
int pathstack[M]; /\*路径栈，存储路径信息\*/
  
int top; /\*栈顶\*/
  
int visited[M]; /\*入栈标记，防止形成回路\*/
  
int count; /\*路径计数器\*/
  
void Dfs(matrix \*g, int s, int e) {
  
 int dis = 0; /\*用于记录路径长度\*/
  
 pathstack[top] = s; /\*将本趟起点入栈\*/
  
 top++;
  
 visited[s] = 1;
  
 int i; /\*将入栈点标记为已入栈\*/
  
 for ( i = 0; i < g->vexNum; i++) {
  
 if (g->edge[s][i] > 0 && g->edge[s][i] != INFINITY && !visited[i]) {
  
 /\*表明前一个入栈点与该点可达，且该点未入栈（未被访问）\*/
  
 if (i == e) { /\*如果深度遍历搜到了终点，就输出刚才的路径\*/
  
 if(count==4){
  
 return;
  
 }
  
 printf("第%d条路:", count++);
  
 int j;
  
 for ( j = 0; j < top; j++) {
  
 printf("%s->", g->vexs[pathstack[j]].name);
  
 if (j < top - 1)
  
 dis = dis + g->edge[pathstack[j]][pathstack[j + 1]]; /\*统计路径长度\*/
  
 }
  
 dis = dis + g->edge[pathstack[top - 1]][e]; /\*最后一条路单独出来，因为e不能入栈\*/
  
 printf("%s\n", g->vexs[e].name);
  
 printf("总长度是：%dm\n\n", dis);
  
 }
  
 else {
  
 Dfs(g, i, e); /\*如果该点不是终点,接着深度搜索\*/
  
 top--; /\*支路全被访问一遍后，顶点出栈\*/
  
 visited[i] = 0; /\*将出栈点标记为已出栈，允许下次访问\*/
  
 }
  
 }
  
 }
  
   
}
  
   
/\*查询任意两个景点之间的所有路径并打印\*/
  
void Allpath\_Print(matrix \*g) {
  
 int s, e;
  
 count = 1; /\*路径计数器\*/
  
 top = 0; /\*栈顶\*/
  
 memset(pathstack, 0, sizeof(pathstack)); /\*路径栈初始化\*/
  
 memset(visited, 0, sizeof(visited)); /\*入栈标记初始化\*/
  
 do {
  
 printf("\n请输入起点编号：");
  
 scanf("%d", &s);
  
 } while (Judge(s));
  
 do {
  
 printf("\n请输入终点编号：");
  
 scanf("%d", &e);
  
 } while (Judge(e));
  
 printf("\n");
  
 Dfs(g, s - 1, e - 1);
  
}
  
   
/\*多景点间求最佳路径\*/
  
void BestPath(matrix \*g) {
  
 int vNum[M] = { 0 }, j = 1; /\*记录用户输入的编号信息\*/
  
 int d = 0; /\*统计全程总长\*/
  
 printf("\n请输入你要游览的第%d个景点的编号（输入-1结束输入）：", j);
  
 scanf("%d", &vNum[j - 1]);
  
 while (vNum[j - 1] != -1 && j < 12) {
  
 while (Judge(vNum[j - 1])) {
  
 printf("\n请输入你要游览的第%d个景点编号：", j);
  
 scanf("%d", &vNum[j - 1]);
  
 }
  
 if (vNum[j - 1] == -1) break;
  
 printf("\n请输入你要游览的第%d个景点编号：", ++j);
  
 scanf("%d", &vNum[j - 1]);
  
 }
  
 printf("\n这是最佳访问路径：");
  
 int i=0;
  
 for ( i = 0; vNum[i] > 0 && vNum[i + 1] > 0; i++) {
  
 printf("%s->", g->vexs[vNum[i] -1].name); /\*输出路径上的起点\*/
  
 Floydprint(g, vNum[i]-1 , vNum[i + 1]-1 ); /\*利用Floyd算法\*/
  
 d += dist[vNum[i] ][vNum[i + 1] -1];
  
 }
  
 printf("%s\n\n", g->vexs[vNum[j - 2] - 1].name); /\*输出路径上的终点\*/
  
 printf("全程总长为：%d\n\n", d);
  
}
  
   
/\*主菜单\*/
  
 int menu() {
  
 int s;
  
 printf("\t\t│ ┌--------------------------------------┐ \n");
  
 printf("\t\t┌--------------├ 欢迎使用广东海洋大学校园导游咨询系统 ┤-----------------┐\n");
  
 printf("\t\t│ └--------------------------------------┘ \t│\n");
  
 printf("\t\t│ \t│\n");
  
 printf("\t\t│ \t│\n");
  
 printf("\t\t│ ┌------------------------┐ ┌------------------------┐ \t│\n");
  
 printf("\t\t│ │1.海大校园景点信息查询 │ │ 2.两景点间最短路径查询 │ \t│\n");
  
 printf("\t\t│ └------------------------┘ └------------------------┘ \t│\n");
  
 printf("\t\t│ \t│\n");
  
 printf("\t\t│ ┌------------------------┐ ┌------------------------┐ \t│\n");
  
 printf("\t\t│ │ 3.多景点间访问路线查询 │ │ 4.公告栏 │ \t│\n");
  
 printf("\t\t│ └------------------------┘ └------------------------┘ \t│\n");
  
 printf("\t\t│ \t│\n");
  
 printf("\t\t│ ┌------------------------┐ ┌------------------------┐ \t│\n");
  
 printf("\t\t│ │ 5.两点之间的三条路径 │ │ 6.退出校园导游咨询系统 │ \t│\n");
  
 printf("\t\t│ └------------------------┘ └------------------------┘ \t│\n");
  
 printf("\t\t│ \t│\n");
  
 printf("\t\t└----------------------------------------------------------------------\t┘\n\n");
  
 printf("\t\t请根据你的需求选择操作：");
  
 scanf("%d", &s);
  
 printf("\n\n");
  
 return s;
  
}
  
   
/\*景点管理\*/
  
struct
  
{
  
 int close; //景点关闭的标记
  
 char name[20]; //存放景点的名字
  
 char reason[100]; //存放景点关闭的原因
  
}scenic\_Manager[M];
  
   
void scenicManager(matrix \*g)
  
{
  
 int n, i, no;
  
 system("cls"); Dis\_Map();
  
 int num;
  
 printf("请输入你要关闭的景点数量：");
  
 scanf("%d",&num);
  
 for(i=0;i<num;i++){
  
 printf("请输入要关闭的景点编号: ");
  
 scanf("%d", &no);
  
 if (no < 1 || no > 36)
  
 {
  
 printf("输入的编号不正确，请重新输入\n");
  
 }
  
 else
  
 {
  
 scenic\_Manager[no].close = INFINITY;
  
 strcpy(scenic\_Manager[no].name, g->vexs[no].name);
  
 printf("\n");
  
 printf("请输入景点关闭的原因: \n");
  
 scanf("%s", scenic\_Manager[no].reason);
  
 printf("景点关闭成功！\n");
  
 }
  
 }
  
   
}
  
   
/\*判断输入的编号是否合理\*/
  
int Judge(int num)
  
{
  
 int i = 0;
  
 if (num == -1)
  
 return i;
  
 if (num < 1 || num>36)
  
 {
  
 printf("\n输入的编号有误，请输入1-36之间的数字！\n");
  
 i = 1;
  
 }
  
 else if(scenic\_Manager[num].close == INFINITY)
  
 {
  
 printf("\n该景点暂时关闭，无法游览。\n");
  
 printf("\n");
  
 printf("关闭原因: %s\n", scenic\_Manager[num].reason);
  
 i = 1;
  
 }
  
   
 return i;
  
}
  
   
/\*管理员界面\*/
  
void Manager()
  
{
  
 int n, num, no, i;
  
 matrix \*g;
  
 int quit = 0;
  
 FILE \* rf;
  
 rf = fopen("Information.txt", "a");
  
 char info[200];
  
 printf("\t\t ┌------------┐ ┌------------┐ ┌------------┐ \n");
  
 printf("\t\t │ 1.发布公告 │ │ 2.景点管理 │ │ 3.退出 │ \n");
  
 printf("\t\t └------------┘ └------------┘ └------------┘ \n");
  
 printf("请输入要选择的编号:\n");
  
 scanf("%d", &n);
  
 switch (n)
  
 {
  
 case 1:
  
 if (rf)
  
 {
  
 printf("输入要发布的公告信息数目: ");
  
 scanf("%d", &no);
  
 printf("\n");
  
 for (i = 1; i <= no; i++)
  
 {
  
 printf("请输入第%d条公告信息编号: ", i);
  
 scanf("%d", &num);
  
 //fputc(num,rf);
  
 printf("\n");
  
 printf("请输入第%d条公告信息内容: ", i);
  
 scanf("%s", info);
  
 //fputs(info,rf);
  
 printf("\n");
  
 fprintf(rf, "%d\n%s\n",num,info);
  
 fflush(rf);
  
 printf("公告发布成功!\n\n");
  
 printf("\n");
  
 }
  
 }
  
 fclose(rf);
  
 Manager();
  
 break;
  
 case 2:
  
 scenicManager(&g);
  
 break;
  
 case 3:
  
 break;
  
 default:
  
 printf("\t\t\t\t\t错误！没有该选项对应的操作。\n\n");
  
 break;
  
 }
  
}
  
   
/\*用户登录\*/
  
int Login()
  
{
  
 int n = 1, num;
  
 char count[20] = "A212";
  
 char password[20] = "212";
  
 char count\_input[20], password\_input[20];
  
 printf("请输入账号:");
  
 scanf("%s", count\_input);
  
 printf("\n");
  
 printf("请输入密码:");
  
 scanf("%s", password\_input);
  
 printf("\n");
  
 if ((strcmp(count, count\_input) == 0) && (strcmp(password, password\_input) == 0))
  
 {
  
 printf("登录成功!\n");
  
 return n;
  
 }
  
 else printf("输入的账号或密码有误,请重新输入!\n\n");
  
 Login();
  
}
  
   
/\*海大大学校园导游咨询系统登录系统\*/
  
int Login\_system()
  
{
  
 int num = 0;
  
 printf("\t\t ┌---------------------------------------┐ \n");
  
 printf("\t\t ├ 欢迎使用广东海洋大学校园导游咨询系统 ┤ \n");
  
 printf("\t\t └---------------------------------------┘ \n");
  
 printf("\t\t 请登录 \n");
  
 printf("\t\t ┌----------------------┐ ┌--------------------------┐ ┌----------------------┐ \n");
  
 printf("\t\t │ 1.管理员登录 │ │2.游客(其他任意键进入) │ │ 3.退出 │ \n");
  
 printf("\t\t └----------------------┘ └--------------------------┘ └----------------------┘ \n");
  
 printf("请输入要选择的编号:\n");
  
 scanf("%d", &num);
  
 switch (num)
  
 {
  
 case 1:
  
 system("cls"); Login(); Manager();system("cls"); Login\_system(); break;
  
 case 2:
  
 system("cls") ;break;
  
 case 3:
  
 system("cls");Exit();return 0;break;
  
 default:
  
; return 0;
  
 break;
  
 }
  
 return 1;
  
}
  
   
/\*公告栏\*/
  
void Information\_print()
  
{
  
 FILE \*rf;
  
 char a[100];
  
 int n,i;
  
 rf = fopen("Information.txt", "r");
  
 if (rf)
  
 {
  
 printf("公告内容如下:\n\n");
  
 while(!feof(rf) )//判断文件是否结束
  
 {
  
 int ch;
  
 if( (ch = fgetc(rf)) != EOF ) //读取的字符正常就输出
  
 putchar(ch);
  
 }
  
 }
  
 fclose(rf);
  
}
  
   
/\*退出菜单栏\*/
  
void Exit() {
  
 printf("\t\t ┌----------------------------------------┐ \n");
  
 printf("\t\t ├ 欢迎下次使用，谢谢！ ┤ \n");
  
 printf("\t\t └----------------------------------------┘ \n");
  
 printf("\t\t \* \* \* \* \* \* \* \* \* \n");
  
 printf("\t\t \* \* \* \n");
  
 printf("\t\t \* \* \* \* \* \* \* \* \* \n");
  
 printf("\t\t \* \* \* \n");
  
 printf("\t\t \* \* \* \* \* \* \* \* \* \n");
  
}
  
   
/\*主函数\*/
  
void main() {
  
 int quit = 0;
  
 matrix g;
  
 Createmap(&g); /\*从文件读取信息建立图\*/
  
 shortPath(&g); /\*Floyd求出dist与path\*/
  
 while(Login\_system()){
  
 while (!quit) { /\*系统退出条件满足判定\*/
  
 switch (menu()) { /\*打印主菜单\*/
  
 case 1:system("cls"); Dis\_Map();searchinfo(&g); break; /\*中北校园景点信息查询\*/
  
 case 2:system("cls"); Dis\_Map();shortpath\_Print(&g); break; /\*两景点间最短路径查询\*/
  
 case 3:system("cls"); Dis\_Map();BestPath(&g); break; /\*多景点间访问路线查询\*/
  
 case 4:system("cls"); Dis\_Map();Information\_print(); break; /\*公告栏查看\*/
  
 case 6:system("cls");quit=1; Login\_system();break; /\*退出校园导游咨询系统\*/
  
 case 5: system("cls"); Dis\_Map(); Allpath\_Print(&g);break;
  
 default:printf("\t\t\t\t\t错误！没有该选项对应的操作。\n\n");
  
 }
  
 system("pause");
  
 system("cls"); /\*清屏\*/
  
 }
  
 }
  
}