

```
/*
Họ và tên : Nguyễn Văn Tèo
Lớp      : 12DHTHxy
Buổi/Tiết : ST6/12345
*/

#include<conio.h>
#include<stdio.h>
#include <iostream>
using namespace std;

#define VC 99
#define MAXE 100
#define MAXV 11
#define TRUE 1
#define FALSE 0

bool visited[MAXV]; //danh dau cac dinh da duoc tham
//=====
struct Graph
{
    int flag; //0: ĐT vô hướng, 1: ĐT có hướng, -1: ĐT không hợp lệ
    int n;    //Số đỉnh
    int w[MAXV][MAXV];
};
//=====
void initGraph(Graph &g)
{
    g.flag = -1;
    g.n = 0;
    for (int i = 0; i <= MAXV; i++)
    {
        for (int j = 0; j <= MAXV; j++)
            g.w[i][j] = VC;
    }
}
//=====
void showGraph(Graph g)
{
    if (g.flag == -1)
    {
        printf("\nKhong phai do thi.");
        getch();
        return;
    }
    printf("\nSo dinh: %d", g.n);
    if (g.flag == 0)
        printf("\nDo thi vo huong");
    else if (g.flag == 1)
        printf("\nDo thi co huong");
    printf("\nDANH SACH CANH:\n");

    printf("    ");
    for (int i = 1; i <= g.n; i++)
        printf("%5d", i);
    printf("\n");

    for (int i = 1; i <= g.n; i++)
    {
        printf("%5d", i);
```

```

        for (int j = 1; j <= g.n; j++)
        {
            printf("%5d", g.w[i][j]);
        }
        printf("\n"); //Xuống dòng
    }
}

//=====
bool read_Adjacent_Matrix_Data(char fileName[], Graph &g)
{ //Đọc Ma trận kề Hoặc Ma trận trọng số
    initGraph(g);
    FILE *fi = fopen(fileName, "rt"); //Read File input - fi
    if (fi == NULL)
    {
        printf("Khong the mo duoc file!");
        getch();
        return false;
    }
    fscanf(fi, "%d\n", &g.flag); //Đọc loại Đồ thị
    fscanf(fi, "%d\n", &g.n); //Đọc số đỉnh
    for (int i = 1; i <= g.n; i++)
    {
        for (int j = 1; j <= g.n; j++)
            fscanf(fi, "%d", &g.w[i][j]); //Đọc số cạnh
    }
    fclose(fi);
    return true;
}

//=====
void input_Start_End(Graph g, int &start, int &end)
{
    int a, b;
    a = b = 0;
    printf("Cac dinh danh so tu 1 den %d.\n", g.n);
    do
    {
        printf("Nhap dinh bat dau: ");
        scanf("%d", &a);
        if (a < 1 || a > g.n)
            printf("Khong hop le! \nNhap lai dinh bat dau: ");
    } while (a < 1 || a > g.n);

    do
    {
        printf("Nhap dinh ket thuc: ");
        scanf("%d", &b);
        if (b < 1 || b > g.n)
            printf("Khong hop le! \nNhap lai dinh ket thuc: ");
    } while (b < 1 || b > g.n);
    start = a;
    end = b;
}

//=====
int Dijkstra(Graph g, int *P, int s, int e)
{
    //int a = s - 1, b = e - 1;
    int a = s, b = e;
    // Len[i] - Gia tri nho nhat tu a -> i. Len1 danh dau do dai.
    int *Len = new int[g.n + 1];
    int *S = new int[g.n + 1]; //Danh dau dinh thuoc danh sach dac biet

```

```
P = new int[g.n + 1]; //truy vet;
```

```
fill(Len, Len + g.n + 1, VC); //Gan duong di ban dau = vo cung
fill(P, P + g.n + 1, a);
fill(S, S + g.n + 1, 0); //Danh sach dac biet
Len[a] = 0; // khoi tao do dai tu a->a = 0
int i = a;
```

```
//while S<>V
for (int k = 1; k <= g.n; k++)
{
    //tim do dai ngan nhat trong cac dinh
    for (i = 1; i <= g.n; i++) // tim v thuoc (V-S) va Len[v] < vo cung
    {
        if (!S[i] && Len[i] != VC)
            break;
    }
    for (int j = i + 1; j <= g.n; j++) // tim dinh co Len min
    {
        if (!S[j] && Len[j] < Len[i])
            i = j;
    }
    S[i] = 1;
```

```
//-----Tinh do dai tu dinh dang xet toi cac dinh tiep
for (int j = 1; j <= g.n; j++) //thay doi do dai neu co
{
    if (!S[j] && g.w[i][j])
    {
        if (Len[i] + g.w[i][j] < Len[j])
        {
            Len[j] = Len[i] + g.w[i][j];
            P[j] = i; //truy vet
        }
    }
}
return Len[b];
```

```
//=====
void main()
{
    Graph G1, G2;
    //Ma trận kề (Ma trận trọng số)
    read_Adjacent_Matrix_Data("input_mtk_01.txt", G1);
    showGraph(G1);
```

```
printf("\n----Thuat toan Dijkstra----\n");
int start, end;
input_Start_End(G1, start, end);
int *P;
int len = Dijkstra(G1, P, start, end);
```

```
// in ket qua
printf("\nDo dai ngan nhat cua duong di tu %d den %d la: %d \n", start, end, len);
printf("Qua trinh duong di: ");
```

```
int i = end;
printf("%d", i);
while (i != start)
```

```
{  
    printf(" <-- ");  
    printf("%d", P[i]);  
    i = P[i];  
}  
getch();  
}
```