#include <iostream>

using namespace std;

#define MaxInt 32767

#define MVNum 100

typedef struct

{

char vexs[MVNum];

int arcs [MVNum][MVNum];

int vexnum,arcnum;

}AMGraph;

//确定顶点在G中的位置

int LocateVex(AMGraph G,char v)

{

int i;

for(i=0;i<G.vexnum;i++)

{

if(G.vexs[i]==v)break;

}

return i;

}

//采用邻接矩阵表示法，创建无向网G

int CreateUDN(AMGraph &G)

{

int i,j,k;int w;

char v1,v2;

cout<<"请输入有向图的总顶点数及边数（以空格隔开）："<<endl;

cin>>G.vexnum>>G.arcnum;

cout<<"请依次输入各顶点的信息（如 a）："<<endl;

for(i=0;i<G.vexnum;++i)

{

cout<<"请依次输入第"<<i+1<<"个顶点：";

cin>>G.vexs[i];

}

for(i=0;i<G.vexnum;++i)

for(j=0;j<G.vexnum;++j)

{

G.arcs[i][j]=MaxInt;

}

cout<<"请输入各条边依附的顶点及权值（如 a b 5）："<<endl;

for(k=0;k<G.arcnum;++k)

{

cout<<"请输入第"<<k+1<<"条边依附的顶点及权值：";

cin>>v1>>v2>>w;

i=LocateVex(G,v1);j=LocateVex(G,v2);

G.arcs[i][j]=w;

G.arcs[j][i]=G.arcs[i][j];

}

return 1;

}

int main()

{

int i,j;

cout << "\*\*\*\*\*\*\*\*\*\*\*\*采用邻接矩阵表示法创建无向网\*\*\*\*\*\*\*\*\*\*\*\*\*\*" <<endl;

AMGraph G;

CreateUDN(G);

cout << "\*\*\*\*\*邻接矩阵表示法创建的无向网\*\*\*\*\*" << endl;

for(i = 0 ; i < G.vexnum ; ++i)

{

for(j = 0; j < G.vexnum; ++j)

{

if(G.arcs[i][j] != MaxInt)

cout << G.arcs[i][j] << "\t";

else

cout << "∞" << "\t";

}

cout<<endl<<endl;

}

cout <<endl;

return 0;

}

无向网有多个父节点和子节点，通过结构来链接各个节点，按照题目要求创建各节点之间的联系