% \*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*  
%  
%          dijkstra.m  
%  
%  The program computes the shortest path in a network  
%  using Dijkstra's algorithm.  
%  
%  The example and the Algorithm is taken from   
%  Worboys, M. F., and M. Duckham, 2004.   
%  GIS: A Computing Perspective, 2nd edition. Taylor & Francis.  
%  Pages: 214-217  
%        
%  Author: Wagaye Jenbere  
%  Date:02/22/2008  
%  
% \*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*  
%  
clear, clf

format short

%

% Specify the number of Vertices in the network

Vertices=8;

%

% Create the adjacency matrix

Adj\_Matrix=[ 0 20 0 0 0 0 15 0 ; ...

20 0 8 9 0 0 0 0 ; ...

0 8 0 6 15 0 0 10 ; ...

0 9 6 0 7 0 0 0 ; ...

0 0 15 7 0 22 18 0 ; ...

0 0 0 0 22 0 0 0 ; ...

15 0 0 0 18 0 0 0 ; ...

0 0 10 0 0 0 0 0 ];

%

% Create the state matrix

State\_Matrix=[ Inf 0 0 ; ...

Inf 0 0 ; ...

Inf 0 0 ; ...

Inf 0 0 ; ...

Inf 0 0 ; ...

Inf 0 0 ; ...

Inf 0 0 ; ...

Inf 0 0 ];

%

% Define the coordinates of the vertices. Be aware of that

% these coordinates are only used for presentation purposes.

% All distances in the computations are based on the

% adjacency matrix.

Coord\_Matrix=[ 0 5 ; 1 0 ; 5 1 ; 2 4 ; 3 6 ; 0 7 ; 4 8 ; 6 2 ];

%

% Draw the network.

text(Coord\_Matrix(1,1), Coord\_Matrix(1,2), '1'),hold on

text(Coord\_Matrix(2,1), Coord\_Matrix(2,2), '2'),hold on

text(Coord\_Matrix(3,1), Coord\_Matrix(3,2), '3'),hold on

text(Coord\_Matrix(4,1), Coord\_Matrix(4,2), '4'),hold on

text(Coord\_Matrix(5,1), Coord\_Matrix(5,2), '5'),hold on

text(Coord\_Matrix(6,1), Coord\_Matrix(6,2), '6'),hold on

text(Coord\_Matrix(7,1), Coord\_Matrix(7,2), '7'),hold on

text(Coord\_Matrix(8,1), Coord\_Matrix(8,2), '8'),hold on

%

for i=1:Vertices

for j=1:(i-1)

if Adj\_Matrix(i,j)~=0

Graphx=[Coord\_Matrix(i,1) Coord\_Matrix(j,1)];

Graphy=[Coord\_Matrix(i,2) Coord\_Matrix(j,2)];

plot(Graphx,Graphy,':'),hold on

end;

end;

end;

%

axis('equal');

axis('off');

%

% Specify the start and end point for the search

startPoint=input('Give the start point: ');

endPoint=input('Give the end point: ');

% Here you should add the shortest path computations

% according to Dijkstra's algorithm.

% Initializing the Matrix

count=startPoint;

for i=1:Vertices

matrix(i,1)=Adj\_Matrix(startPoint,count);

matrix(i,2)=0;

matrix(i,3)=0;

count=count+1;

if count==Vertices+1

count=1;

end

end

% The end point assignment

dist=abs(startPoint-endPoint);

for i=1:8

if i==dist

tmp=matrix(8,1);

matrix(8,1)=matrix(i+1,1);

matrix(i+1,1)=tmp;

end

end

% The first visited point assignment

matrix(1,3)=1;

matrix(1,2)=startPoint;

% To get the first non-zero value in the inital matrix

for i = 1 : Vertices

if matrix(i,1)~=0

tmp=matrix(i,1);

break;

end

end

% To check smallest value

for i = 1 : Vertices

if matrix(i,1)~=0 & tmp>matrix(i,1)

tmp=matrix(i,1);

end

end

% Setting the path

path=startPoint; % The path has to be checked again

% To get the point where the distance is the smallest

for i=1:Vertices

if tmp==matrix(i,1)

matrix(i,3)=1;

matrix(i,2)=path;

m=i

else

n(i)=i

end

end

disp(i);

disp(matrix)

%  End  
%

% % \*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

%

% Wagaye\_point\_in\_polygon.m

%

% This program plots polygons

% There are given data and the plan is to find out the number of

% a polygon and to compute its area and highlights it when users

% click on it.

%

% Author: Wagaye Jenbere

% Date: 26/02/2008

%

%%\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

%%

clear;

%%\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*Given Data\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

%The coordinates of all the vertices are loaded from cord.txt

cord=[230 400

270 420

300 400

340 430

390 380

230 300

300 290

370 280

250 200

290 200

320 200

200 190

250 100

310 120

410 110

390 180

];

% %The vertices of the different polygons

poly=[1 6 7 3 2 1 0 0 1;...

3 7 8 5 4 3 0 0 3;...

6 9 10 7 6 0 0 0 6;...

7 10 11 8 7 0 0 0 7;...

6 12 13 14 10 9 6 0 6;...

8 11 10 14 15 16 8 0 8];

%

%%

%marking different vertices.

%

text(cord(1,1), cord(1,2), '1' ),hold on

text(cord(2,1), cord(2,2), '2' ),hold on

text(cord(3,1), cord(3,2), '3' ),hold on

text(cord(4,1), cord(4,2), '4' ),hold on

text(cord(5,1), cord(5,2), '5' ),hold on

text(cord(6,1), cord(6,2), '6' ),hold on

text(cord(7,1), cord(7,2), '7' ),hold on

text(cord(8,1), cord(8,2), '8' ),hold on

text(cord(9,1), cord(9,2), '9' ),hold on

text(cord(10,1), cord(10,2), '10' ),hold on

text(cord(11,1), cord(11,2), '11' ),hold on

text(cord(12,1), cord(12,2), '12' ),hold on

text(cord(13,1), cord(13,2), '13' ),hold on

text(cord(14,1), cord(14,2), '14' ),hold on

text(cord(15,1), cord(15,2), '15' ),hold on

text(cord(16,1), cord(16,2), '16' ),hold on;

%

%Plotting the polygons

for i=1:6

for j=1:8

m=poly(i,j);

n=poly(i,j+1);

if n~=0 & m~=0

X=[cord(m,1) cord(n,1)];

Y=[cord(m,2) cord(n,2)];

plot(X,Y), hold on

end;

end;

end;

%

%\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*Response to selecting a polygon by user and computing its area\*\*\*\*\*

%In this section user will click on an arbitrary polygon and the program

%will tell which polygon clicked, and will compute its area.

%To keep the whole programme under a big 'while'loop.

%here a variable named otherpoly was declared.

otherpoly=1;

while otherpoly==1

% Ask user to click on a polygon

disp('Double click on a polygon')

% The point that user click is stored as (O,P)

[O,P]= getpts;

% Then the following programme will check which polygon clicked.

for r=1:6

for s=1:8

t=poly(r,s);

if t~=0

x(s)=cord(t,1);

y(s)=cord(t,2);

end;

end;

% point in polygon function "inpolygon" has been used

In=inpolygon(O,P,x,y);

% if it is inside the value of In will be 1

if In==1

% display the number of the polygon clicked

disp('Number of the polygon is')

disp(r)

%Then compute the area of the selected polygon

area=polyarea(x,y);

%display the area of the polygon

disp('Area of the polygon is')

disp(area)

% Highlighting the selected polygon

for j=1:8

m=poly(r,j);

n=poly(r,j+1);

if m~=0 & n~=0

X=[cord(m,1) cord(n,1)];

Y=[cord(m,2) cord(n,2)];

plot(X,Y,'--b','LineWidth',2.5,'MarkerEdgeColor','k')

end;

end;

end;

end;

% To select other points

otherpoly=input('To continue please press "1 and enter" or other key to stop ');

end;

%

%\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*