# 5) Polygon Filling

#include<iostream>

#include<conio.h>

#include<graphics.h>

using namespace std;

struct edge

{

int x1,y1,x2,y2,flag;

};

int main()

{

initwindow(800,800);

int n;

cout<<"Enter the no.of vertices of the graph :"<<endl;

cin>>n;

struct edge ed[n];

struct edge temped;

float dx,dy,m[n],x\_int[n],inter\_x[n];

int x[n],y[n],ymax=0,ymin=900,yy,temp;

/\*read the vertices of the polygon and also find ymax and ymin\*/

cout<<"Enter the vertices"<<endl;

for(int i=0;i<n;i++)

{

cin>>x[i];

cin>>y[i];

if(y[i]>ymax)

ymax=y[i];

if(y[i]<ymin)

ymin=y[i];

ed[i].x1=x[i];

ed[i].y1=y[i];

}

/\*store the edge information\*/

for(int i=0;i<n-1;i++)

{

ed[i].x2=ed[i+1].x1;

ed[i].y2=ed[i+1].y1;

ed[i].flag=0;

}

ed[n-1].x2=ed[0].x1;

ed[n-1].y2=ed[0].y1;

ed[n-1].flag=0;

/\*Check for y1>y2, if not interchnge y1 and y2 \*/

for(int i=0;i<n;i++)

{

if(ed[i].y1 < ed[i].y2)

{

temp=ed[i].x1;

ed[i].x1=ed[i].x2;

ed[i].x2=temp;

temp=ed[i].y1;

ed[i].y1=ed[i].y2;

ed[i].y2=temp;

}

}

/\*Draw the polygon\*/

for(int i=0;i<n;i++)

{

line(ed[i].x1, ed[i].y1,ed[i].x2,ed[i].y2);

}

/\*sorting of edges in the order of y1,y2,x1,x2\*/

for(int i=0;i<n-1;i++)

{

for(int j=0;j<n-1;j++)

{

if(ed[j].y1<ed[j+1].y1)

{

temped=ed[j];

ed[j]=ed[j+1];

ed[j+1]=temped;

}

if(ed[j].y1==ed[j+1].y1)

{

if(ed[j].y2<ed[j+1].y2)

{

temped=ed[j];

ed[j]=ed[j+1];

ed[j+1]=temped;

}

if (ed[j].y2==ed[j+1].y2)

{

if(ed[j].x1<ed[j+1].x1)

{

temped=ed[j];

ed[j]=ed[j+1];

ed[j+1]=temped;

}

}

}

}

}

/\*calculating 1/slope of each edge and storing x\*/

for(int i=0;i<n;i++)

{

dx=ed[i].x2-ed[i].x1;

dy=ed[i].y2-ed[i].y1;

if(dy==0)

{

m[i]=0;

}

else

{

m[i]=dx/dy;

}

inter\_x[i]=ed[i].x1;

}

/\*making the Actual edges\*/

yy=ymax;

while(yy>ymin)

{

for(int i=0;i<n;i++)

{

if(yy>ed[i].y2 && yy<=ed[i].y1)

{

ed[i].flag=1;

}

else

ed[i].flag=0;

}

int j=0;

for(int i=0;i<n;i++)

{

if(ed[i].flag==1)

{

if(yy==ed[i].y1)

{

x\_int[j]==ed[i].x1;

j++;

if(ed[i-1].y1==yy && ed[i-1].y1<yy)

{

x\_int[j]=ed[i].x1;

j++;

}

if(ed[i+1].y1==yy && ed[i+1].y1<yy)

{

x\_int[j]=ed[i].x1;

j++;

}

}

else

{

x\_int[j]=inter\_x[i]+(-m[i]);

inter\_x[i]=x\_int[j];

j++;

}

}

}

/\*sorting the x intersection\*/

for(int i=0;i<j;i++)

{

for(int k=0;k<j-1;k++)

{

if(x\_int[k]>x\_int[k+1])

{

temp=(int)x\_int[k];

x\_int[k]=x\_int[k+1];

x\_int[k+1]=temp;

}

}

}

/\*extracting pairs of values to draw lines\*/

for(int i=0;i<j;i=i+2)

{

line((int)x\_int[i],yy,(int)x\_int[i+1],yy);

}

yy--;

delay(50);

}

closegraph();

getch();

return 0;

}

**OUTPUT:**

