**/\* Program No. :**

**Aim : WAP for polygon clipping.**

**\*/**

#include<stdio.h>

#include<graphics.h>

#include<conio.h>

#include<string.h>

#define no\_ver 5

int vertices,i,j,minx,miny,maxx,maxy;

int poly[(no\_ver+1)\*2],newpoly[(no\_ver+1)\*2];

float cutx,cuty;

void set\_intersection(char \*clipside,int x1,int y1,int x2,int y2)

{

float slope;

if(x1!=x2)

slope=(y2-y1)/(x2-x1);

if(strcmp(clipside,"left")==0)

{

cutx=minx;

cuty=y1+slope\*(minx-x1);

}

if(strcmp(clipside,"right")==0)

{

cutx=maxx;

cuty=y1+slope\*(maxx-x1);

}

if(strcmp(clipside,"bottom")==0)

{

if(x1!=x2)

cutx=x1+(maxy-y1)/slope;

else

cutx=x1;

cuty=maxy;

}

if(strcmp(clipside,"top")==0)

{

if(x1!=x2)

cutx=x1+(miny-y1)/slope;

else

cutx=x1;

cuty=miny;

}

}

void clippoly(int edges,int poly[])

{

int temppoly[(no\_ver+1)\*2];

int i,counter=0;

for(i=0;i<edges\*2;i+=2)

{

// left clip...

if(poly[i]>minx)

{

temppoly[counter++]=poly[i];

temppoly[counter++]=poly[i+1];

}

else

{

if(i==0)

{

set\_intersection("left",poly[i],poly[i+1],poly[edges\*2-4],poly[edges\*2-3]);

temppoly[counter++]=cutx;

temppoly[counter++]=cuty;

set\_intersection("left",poly[i],poly[i+1],poly[i+2],poly[i+3]);

temppoly[counter++]=cutx;

temppoly[counter++]=cuty;

}

else

{

set\_intersection("left",poly[i-2],poly[i-1],poly[i],poly[i+1]);

temppoly[counter++]=cutx;

temppoly[counter++]=cuty;

set\_intersection("left",poly[i],poly[i+1],poly[i+2],poly[i+3]);

temppoly[counter++]=cutx;

temppoly[counter++]=cuty;

}

}

}

edges=counter/2;

counter=0;

for(i=0;i<edges\*2;i+=2)

{

// right clip...

if(temppoly[i]<maxx)

{

newpoly[counter++]=temppoly[i];

newpoly[counter++]=temppoly[i+1];

}

else

{

if(i==0)

{

set\_intersection("right",temppoly[i],temppoly[i+1],temppoly[edges\*2-4],temppoly[edges\*2-3]);

newpoly[counter++]=cutx;

newpoly[counter++]=cuty;

set\_intersection("right",temppoly[i],temppoly[i+1],temppoly[i+2],temppoly[i+3]);

newpoly[counter++]=cutx;

newpoly[counter++]=cuty;

}

else

{

set\_intersection("right",temppoly[i-2],temppoly[i-1],temppoly[i],temppoly[i+1]);

newpoly[counter++]=cutx;

newpoly[counter++]=cuty;

set\_intersection("right",temppoly[i],temppoly[i+1],temppoly[i+2],temppoly[i+3]);

newpoly[counter++]=cutx;

newpoly[counter++]=cuty;

}

}

}

edges=counter/2;

counter=0;

for(i=0;i<edges\*2;i+=2)

{

// bottom clip...

if(newpoly[i+1]<maxy)

{

temppoly[counter++]=newpoly[i];

temppoly[counter++]=newpoly[i+1];

}

else

{

if(i==0)

{

set\_intersection("bottom",newpoly[i],newpoly[i+1],newpoly[edges\*2-4],newpoly[edges\*2-3]);

temppoly[counter++]=cutx;

temppoly[counter++]=cuty;

set\_intersection("bottom",newpoly[i],newpoly[i+1],newpoly[i+2],newpoly[i+3]);

temppoly[counter++]=cutx;

temppoly[counter++]=cuty;

}

else

{

set\_intersection("bottom",newpoly[i-2],newpoly[i-1],newpoly[i],newpoly[i+1]);

temppoly[counter++]=cutx;

temppoly[counter++]=cuty;

set\_intersection("bottom",newpoly[i],newpoly[i+1],newpoly[i+2],newpoly[i+3]);

temppoly[counter++]=cutx;

temppoly[counter++]=cuty;

}

}

}

edges=counter/2;

counter=0;

for(i=0;i<edges\*2;i+=2)

{

// top clip...

if(temppoly[i+1]>miny)

{

newpoly[counter++]=temppoly[i];

newpoly[counter++]=temppoly[i+1];

}

else

{

if(i==0)

{

set\_intersection("top",temppoly[i],temppoly[i+1],temppoly[edges\*2-4],temppoly[edges\*2-3]);

newpoly[counter++]=cutx;

newpoly[counter++]=cuty;

set\_intersection("top",temppoly[i],temppoly[i+1],temppoly[i+2],temppoly[i+3]);

newpoly[counter++]=cutx;

newpoly[counter++]=cuty;

}

else

{

set\_intersection("top",temppoly[i-2],temppoly[i-1],temppoly[i],temppoly[i+1]);

newpoly[counter++]=cutx;

newpoly[counter++]=cuty;

set\_intersection("top",temppoly[i],temppoly[i+1],temppoly[i+2],temppoly[i+3]);

newpoly[counter++]=cutx;

newpoly[counter++]=cuty;

}

}

}

clrscr();

setcolor(RED);

rectangle(minx,miny,maxx,maxy);

setcolor(GREEN);

drawpoly(counter/2,newpoly);

}

void main()

{

int gdriver = DETECT, gmode, errorcode;

initgraph(&gdriver, &gmode, "C:\\TC3.0\\BGI");

errorcode = graphresult();

clrscr();

if (errorcode != grOk)

{

printf("Graphics error: %s\n", grapherrormsg(errorcode));

printf("Press any key to exit.");

getch();

exit(1);

}

printf("\n\nEnter the number of vertices : ");

scanf("%d",&vertices);

if(vertices>no\_ver)

{

printf("\n\n\tMaximum number of vertices are %d",no\_ver);

printf("\n\tPress any key to exit.");

getch();

exit(1);

}

else

{

for(i=0,j=0;i<vertices;i++,j+=2)

{

printf("\n\nEnter the coordinates of vertice %d : ",i+1);

printf("\n\t\tx : ");

scanf("%d",&poly[j]);

printf("\t\ty : ");

scanf("%d",&poly[j+1]);

}

}

poly[j]=poly[0];

poly[j+1]=poly[1];

printf("\n\nEnter the starting coordinates of the clip window :");

printf("\n\t\tx : ");

scanf("%d",&minx);

printf("\t\ty : ");

scanf("%d",&miny);

printf("\n\nEnter the ending coordinates of the clip window :");

printf("\n\t\tx : ");

scanf("%d",&maxx);

printf("\t\ty : ");

scanf("%d",&maxy);

clrscr();

setcolor(RED);

rectangle(minx,miny,maxx,maxy);

setcolor(GREEN);

drawpoly(vertices+1,poly);

getch();

clippoly(vertices+1,poly);

getch();

closegraph();

}

**/\***

**Name : Rohit Aggarwal**

**Roll No. : 7CS-097**

**\*/**