**LINE CLIPPING USING NICOL-LEE-NICOL ALGORITHM:**

#include<stdio.h>

#include<conio.h>

#include<math.h>

#include <graphics.h>

int xmin,ymin,xmax,ymax;

void main()

{

int x1,y1,x2,y2;

int gdriver = DETECT, gmode, errorcode;

int findRegionP1(int,int);

void clipline1(int,int,int,int);

void clipline2(int,int,int,int);

void clipline3(int,int,int,int);

int ch;

float m;

clrscr();

printf("\nEnter the xmin:->");

scanf("%d",&xmin);

printf("\nEnter the ymin:->");

scanf("%d",&ymin);

printf("\nEnter the xmax:->");

scanf("%d",&xmax);

printf("\nEnter the ymax:->");

scanf("%d",&ymax);

initgraph(&gdriver, &gmode, "C:\\TC\\BGI");

setcolor(15);

rectangle(xmin,ymin,xmax,ymax);

printf("Enter the x1:->");

scanf("%d",&x1);

printf("Enter the y1:->");

scanf("%d",&y1);

printf("Enter the x2:->");

scanf("%d",&x2);

printf("Enter the y2:->");

scanf("%d",&y2);

setcolor(12);

line(x1,y1,xmin,ymin);

line(x1,y1,xmax,ymin);

line(x1,y1,xmax,ymax);

line(x1,y1,xmin,ymax);

getch();

setcolor(3);

line(x1,y1,x2,y2);

getch();

ch=first\_end\_point\_region(x1,y1);

switch(ch)

{

case 1:

clipline1(x1,y1,x2,y2);

break;

case 2:

clipline2(x1,y1,x2,y2);

break;

case 3:

clipline3(x1,y1,x2,y2);

break;

default:

printf("\nInvalid select of the option: ");

}

getch();

}

int first\_end\_point\_region(int x,int y)

{

/\* u have two equations:-xmin <= x <= xmax;ymin <= y <= ymax; \*/

if(x>=xmin && x<=xmax && y>=ymin && y<=ymax)

return 1;

elseif(x<xmin && y>=ymin && y<=ymax)

return 2;

elseif(x<=xmin && y<=ymin)

return 3;

elsereturn 0;

}

/\* point p1 is inside the clip window \*/

void clipline1(int x1,int y1,int x2,int y2)

{

int draw=1;

float m,m1,m2,m3,m4;

int nx1,ny1,nx2,ny2;

/\* calculate slopes for all the lines passing thru vertices and including the input line :- \*/

m=((float)(y2-y1))/(x2-x1);

m1=((float)(ymin-y1))/(xmin-x1);

m2=((float)(ymin-y1))/(xmax-x1);

m3=((float)(ymax-y1))/(xmax-x1);

m4=((float)(ymax-y1))/(xmin-x1);

nx1=x1;

ny1=y1;

// point p2 is on topif(((abs(m)>=m1 && x2<x1) || (abs(m)>abs(m2) && x2>x1)) && y1>y2)

{

// point p2 is also inside clip windowif(y2>ymin)

{

nx2=x2;

ny2=y2;

}

// point p2 is outside clip windowelse

{

ny2=ymin;

nx2=x1+(ymin-y1)/m;

}

}

// point p2 is on right side of clip windowelseif(m>m2 && m<m3 && x2>=xmax)

{

// point p2 is inside clip windowif(x2<xmax)

{

nx2=x2;

ny2=y2;

}

// point p2 is outside clip windowelse

{

nx2=xmax;

ny2=y1+(xmax-x1)\*m;

}

}

// point p2 is on bottom side of clip windowelseif((abs(m)>=m3 && x2>x1) || (abs(m)>abs(m4) && x2<x1))

{

// point p2 is inside clip windowif(y2<ymax)

{

nx2=x2;

ny2=y2;

}

// point p2 is outside clip windowelse

{

ny2=ymax;

nx2=x1+(ymax-y1)/m;

}

}

// point p2 is on left side of clip windowelseif(m>m4 && m<m1)

{

// point p2 is inside the clip windowif(x2>xmin)

{

nx2=x2;

ny2=y2;

}

// point p2 is outside the clip windowelse

{

nx2=xmin;

ny2=y1+(xmin-x1)\*m;

}

}

else

draw=0;

getch();

cleardevice();

setcolor(18);

rectangle(xmin,ymin,xmax,ymax);

if(draw)

{

setcolor(12);

line(x1,y1,xmin,ymin);

line(x1,y1,xmax,ymin);

line(x1,y1,xmax,ymax);

line(x1,y1,xmin,ymax);

setcolor(5);

line(nx1,ny1,nx2,ny2);

}

}

/\* Point p1 is in the edge region \*/void clipline2(int x1,int y1,int x2,int y2)

{

int draw=1;

float m,m1,m2,m3,m4;

int nx1,ny1,nx2,ny2;

m=((float)(y2-y1))/(x2-x1);

m1=((float)(ymin-y1))/(xmin-x1);

m2=((float)(ymin-y1))/(xmax-x1);

m3=((float)(ymax-y1))/(xmax-x1);

m4=((float)(ymax-y1))/(xmin-x1);

// Point p2 is in Left-Top regionif(m>m1 && m<m2)

{

// Point p2 is inside the clip windowif(y2>ymin)

{

nx1=xmin;

ny1=y1+m\*(xmin-x1);

nx2=x2;

ny2=y2;

}

// Point p2 is outside the clip windowelse

{

nx1=xmin;

ny1=y1+m\*(xmin-x1);

ny2=ymin;

nx2=x1+(ymin-y1)/m;

}

}

// Point p2 is in Left-Right regionelseif(m>m2 && m<m3)

{

// Point p2 is inside the clip windowif(x2<xmax)

{

nx1=xmin;

ny1=y1+m\*(xmin-x1);

nx2=x2;

ny2=y2;

}

// Point p2 is outside the clip windowelse

{

nx1=xmin;

ny1=y1+m\*(xmin-x1);

nx2=xmax;

ny2=y1+(xmax-x1)\*m;

}

}

// Point p2 is in Left-Bottom regionelseif(m>m3 && m<m4)

{

// Point p2 is inside the clip windowif(y2<ymax)

{

nx1=xmin;

ny1=y1+m\*(xmin-x1);

nx2=x2;

ny2=y2;

}

// Point p2 is outside the clip windowelse

{

nx1=xmin;

ny1=y1+m\*(xmin-x1);

ny2=ymax;

nx2=x1+(ymax-y1)/m;

}

}

else

draw=0;

getch();

cleardevice();

setcolor(18);

rectangle(xmin,ymin,xmax,ymax);

if(draw)

{

setcolor(12);

line(x1,y1,xmin,ymin);

line(x1,y1,xmax,ymin);

line(x1,y1,xmax,ymax);

line(x1,y1,xmin,ymax);

setcolor(5);

line(nx1,ny1,nx2,ny2);

}

}

/\* Point p1 is in the Corner Region \*/

void clipline3(int x1,int y1,int x2,int y2)

{

int draw=1;

float m,m1,m2,m3,m4,tm1,tm2;

int nx1,ny1,nx2,ny2;

int flag,t;

tm1=((float)(ymin-y1))/(xmin-x1);

tm2=((float)(ymax-ymin))/(xmax-xmin); //diagonal slope

m=((float)(y2-y1))/(x2-x1);

m1=((float)(ymin-y1))/(xmax-x1);

m2=((float)(ymax-y1))/(xmax-x1);

m3=((float)(ymin-y1))/(xmin-x1);

m4=((float)(ymax-y1))/(xmin-x1);

// Point p1 is towards the left side of the clip window (case2)if(tm1<tm2)

{

flag=2;

t=m2;

m2=m3;

m3=t;

}

// Point p1 is towards the top side of the clip window (case1)else

flag=1;

// Point p2 is in the Top-Right regionif(m>m1 && m<m2)

{

// Point p2 is outside the clip windowif(x2>xmax && y2>ymin)

{

ny1=ymin;

nx1=x1+(ymin-y1)/m;

nx2=xmax;

ny2=y1+m\*(xmax-x1);

}

// Point p2 is inside the clip windowelseif(y2>ymin && x2<xmax)

{

ny1=ymin;

nx1=x1+(ymin-y1)/m;

ny2=y2;

nx2=x2;

}

}

// Point p2 is Left-Right or Top-Bottom regionelseif(m>m2 && m<m3)

{

// Point p2 is in Top-Bottom region (case1)if(flag==1)

{

// Point p2 is outside the clip windowif(y2>=ymax)

{

ny1=ymin;

nx1=x1+(ymin-y1)/m;

nx2=x1+(ymax-y1)/m;

ny2=ymax;

}

// Point p2 is inside the clip windowelseif(y2>=ymin)

{

ny1=ymin;

nx1=x1+(ymin-y1)/m;

nx2=x2;

ny2=y2;

}

}

// Point p2 is in Left-Right region (case2)else

{

// Point p2 is outside the clip windowif(x2>=xmax)

{

nx1=xmin;

ny1=y1+m\*(xmin-x1);

nx2=xmax;

ny2=y1+m\*(xmax-x1);

}

// Point p2 is inside the clip windowelseif(x2>=xmin)

{

nx1=xmin;

ny1=y1+m\*(xmin-x1);

nx2=x2;

ny2=y2;

}

}

}

// Point p2 is in Left-Bottom regionelseif(m>m3 && m<m4)

{

// Point p2 is outside the clip windowif(y2>=ymax)

{

nx1=xmin;

ny1=y1+m\*(xmin-x1);

nx2=x1+(ymax-y1)/m;

ny2=ymax;

}

// Point p2 is inside the clip windowelseif(y2>=ymin)

{

nx1=xmin;

ny1=y1+m\*(xmin-x1);

ny2=y2;

nx2=x2;

}

}

else

draw=0;

getch();

cleardevice();

setcolor(18);

rectangle(xmin,ymin,xmax,ymax);

if(draw)

{

setcolor(12);

line(x1,y1,xmin,ymin);

line(x1,y1,xmax,ymin);

line(x1,y1,xmax,ymax);

line(x1,y1,xmin,ymax);

setcolor(5);

line(nx1,ny1,nx2,ny2);

}

}