**Write a program for implementation of line clipping algorithm.**

#include<iostream.h>

#include<conio.h>

#include<dos.h>

#include<graphics.h>

#include<stdlib.h>

#include<math.h>

#include<stdio.h>

int i,j,pixel[2][4];

float xn1,xn2,yn1,yn2,x3,y3,n,m;

void showquad()

{

cleardevice();

rectangle(120,40,320,240);

rectangle(320,40,520,240);

rectangle(120,240,320,440);

rectangle(320,240,520,440);

for(i=130;i<510;i+=10)

for(j=50;j<430;j+=10)

putpixel(i,j,15);

for(i=130;i<510;i+=10)

{

if(i==320)

continue;

outtextxy(i,237,"+");

}

for(i=50;i<430;i+=10)

{

if(i==240)

continue;

outtextxy(317,i,"-");

}

outtextxy(310,230,"O");

outtextxy(530,240,"X");

outtextxy(320,450,"Y");

outtextxy(100,240,"-X");

outtextxy(320,30,"-Y");

}

void su\_cu(int x1,int y1,int x2,int y2,int xmin,int ymin,int xmax,int ymax)

{

int i,j,f1;

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

for(j=0;j<4;j++)

pixel[i][j]=0;

if(y1>ymax)

pixel[0][0]=1;

if(y1<ymin)

pixel[0][1]=1;

if(x1>xmax)

pixel[0][2]=1;

if(x1<xmin)

pixel[0][3]=1;

if(y2>ymax)

pixel[1][0]=1;

if(y2<ymin)

pixel[1][1]=1;

if(x2>xmax)

pixel[1][2]=1;

if(x2<xmin)

pixel[1][3]=1;

for(j=0;j<4;j++)

{

if(pixel[0][j]==0 && pixel[1][j]==0)

continue;

if(pixel[0][j]==1 && pixel[1][j]==1)

{

f1=3;

break;

}

f1=2;

}

switch(f1)

{

case 1:

line(320+x1,240-y1,320+x2,240-y2);

break;

case 3:

cout<<"Line is not visible";

break;

case 2:

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

xn1=x1;

yn1=y1;

xn2=x2;

yn2=y2;

if(pixel[0][0]==1)

{

xn1=x1+(ymax-y1)/m;

yn1=ymax;

}

if(pixel[0][1]==1)

{

xn1=x1+(ymin-y1)/m;

yn1=ymin;

}

if(pixel[0][2]==1)

{

yn1=y1+(xmax-x1)\*m;

xn1=xmax;

}

if(pixel[0][3]==1)

{

yn1=y1+(xmin-x1)\*m;

xn1=xmin;

}

if(pixel[1][0]==1)

{

xn2=x2+(ymax-y2)/m;

yn2=ymax;

}

if(pixel[1][1]==1)

{

xn2=x2+(ymin-y2)/m;

yn2=ymin;

}

if(pixel[1][2]==1)

{

yn2=y2+(xmax-x2)\*m;

xn2=xmax;

}

if(pixel[1][3]==1)

{

yn2=y2+(xmin-x2)\*m;

xn2=xmin;

}

line(320+xn1,240-yn1,320+xn2,240-yn2);

break;

}

}

void show\_message()

{

char \*mess[]={"-","=","[",",","L","i","n","e","c","l","i","p","p","i","n","g",",","]","=","-"};

int xx=29,xxx=50,i,j;

\_setcursortype(\_NOCURSOR);

for(i=0,j=21;i<13,j>=11;i++,j--)

{

gotoxy(xx,1);

cout<<mess[i];

xx++;

gotoxy(xxx,1);

cout<<mess[j];

xxx--;

delay(50);

}

\_setcursortype(\_NORMALCURSOR);

}

void main()

{

clrscr();

int gd=DETECT,gm,i,j;

int xmin,ymin,xmax,ymax,x1,y1,x2,y2;

int choice,ed[20],num;

show\_message();

cout<<"\n\n\t Enter the cordinate of line :";

cout<<"\nEnter value of x1 and y1 :";

cin>>x1>>y1;

cout<<"\nEnter value of x2 and y2 :";

cin>>x2>>y2;

cout<<"\n Enter the coordinates of the clipping window";

cout<<"\n Enterxmin & ymin: ";

cin>>xmin>>ymin;

cout<<"\n Enter xmax & ymax: ";

cin>>xmax>>ymax;

clrscr();

initgraph(&gd,&gm,"c:\\turboc3\\bgi");

cleardevice();

showquad();

line(320+xmin,240-ymin,320+xmin,240-ymax);

line(320+xmin,240-ymax,320+xmax ,240-ymax);

line(320+xmax,240-ymax,320+xmax,240-ymin);

line(320+xmax,240-ymin,320+xmin,240-ymin);

line(320+x1,240-y1,320+x2,240-y2);

getch();

cleardevice();

showquad();

line(320+xmin,240-ymin,320+xmin,240-ymax);

line(320+xmin,240-ymax,320+xmax ,240-ymax);

line(320+xmax,240-ymax,320+xmax,240-ymin);

line(320+xmax,240-ymin,320+xmin,240-ymin);

su\_cu(x1,y1,x2,y2,xmin,ymin,xmax,ymax);

getch();

exit(0);

cleardevice();

closegraph();

}





