**PROGRAM NO. 11**

**AIM-** To clip a line intersecting at one point withgiven window using Cohen Sutherland LineClipping algorithm.

#include<iostream.h>

#include<conio.h>

#include<math.h>

#include<graphics.h>

#include<process.h>

int pixels[2][4];

float xn1,xn2,yn1,yn2,m;

void cohen(float x1,float y1,float x2,float y2,float xmin,float ymin,float xmax,float ymax)

{

int i,j,fl;

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

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

pixels[i][j]=0;

if(y1>ymax)

pixels[0][0]=1;

if(y1<ymin)

pixels[0][1]=1;

if(x1>xmax)

pixels[0][2]=1;

if(x1<xmin)

pixels[0][3]=1;

if(y2>ymax)

pixels[1][0]=1;

if(y2<ymin)

pixels[1][1]=1;

if(x2>xmax)

pixels[1][2]=1;

if(x2<xmin)

pixels[1][3]=1;

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

{

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

fl=1;

else if(pixels[0][j]==1&&pixels[1][j]==1)

{

fl=3;

break;

}

else

fl=2;

}

switch(fl)

{

case 1:

cout<<"line is visible";

break;

case 3:

cout<<"Line Is Not Visible";

break;

case 2:

cout<<"CLipping Candidate";

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

xn1=x1;

yn1=y1;

xn2=x2;

yn2=y2;

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

{

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

yn1=ymax;

}

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

{

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

yn1=ymin;

}

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

{

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

xn1=xmax;

}

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

{

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

xn1=xmin;

}

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

{

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

yn2=ymax;

}

if(pixels[1][1]==1)

{

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

yn2=ymin;

}

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

{

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

xn2=xmax;

}

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

{

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

xn2=xmin;

}

line(xn1,yn1,xn2,yn2);

cout<<"Line Clipped";

break;

}

}

void main()

{

int gd=DETECT,gm,i,j;

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

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

clearviewport();

cout<<"Enter xmin,ymin,xmax,ymax \n";

cin>>xmin>>ymin>>xmax>>ymax;

rectangle(xmin,ymin,xmax,ymax);

cout<<"Enter x1,y1,x2,y2 \n";

cin>>x1>>y1>>x2>>y2;

line(x1,y1,x2,y2);

setcolor(YELLOW);

cohen(x1,y1,x2,y2,xmin,ymin,xmax,ymax);

getch();

closegraph();

}

**OUTPUT:-**

