***LINE CLIPPING USING LIANG BARSKY :***

#include<stdio.h>

#include<graphics.h>

#include<math.h>

#include<dos.h>

void main()

{

int i,gd=DETECT,gm;

int x1,y1,x2,y2,xmin,xmax,ymin,ymax,xx1,xx2,yy1,yy2,dx,dy;

float t1,t2,p[4],q[4],temp;

x1=120;

y1=120;

x2=300;

y2=300;

xmin=100;

ymin=100;

xmax=250;

ymax=250;

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

rectangle(xmin,ymin,xmax,ymax);

dx=x2-x1;

dy=y2-y1;

p[0]=-dx;

p[1]=dx;

p[2]=-dy;

p[3]=dy;

q[0]=x1-xmin;

q[1]=xmax-x1;

q[2]=y1-ymin;

q[3]=ymax-y1;

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

{

if(p[i]==0)

{

printf("line is parallel to one of the clipping boundary");

if(q[i]>=0)

{

if(i<2)

{

if(y1<ymin)

y1=ymin;

if(y2>ymax)

y2=ymax;

line(x1,y1,x2,y2);

}

if(i>1)

{

if(x1<xmin)

x1=xmin;

if(x2>xmax)

x2=xmax;

line(x1,y1,x2,y2);

}}} }

t1=0;

t2=1;

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

{

temp=q[i]/p[i];

if(p[i]<0)

{

if(t1<=temp)

t1=temp;

}

else

{

if(t2>temp)

t2=temp;

}

}

if(t1<t2)

{

xx1 = x1 + t1 \* p[1];

xx2 = x1 + t2 \* p[1];

yy1 = y1 + t1 \* p[3];

yy2 = y1 + t2 \* p[3];

line(xx1,yy1,xx2,yy2);

}

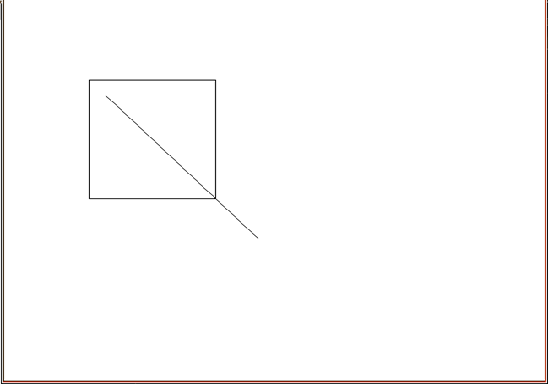
delay(5000);

closegraph();

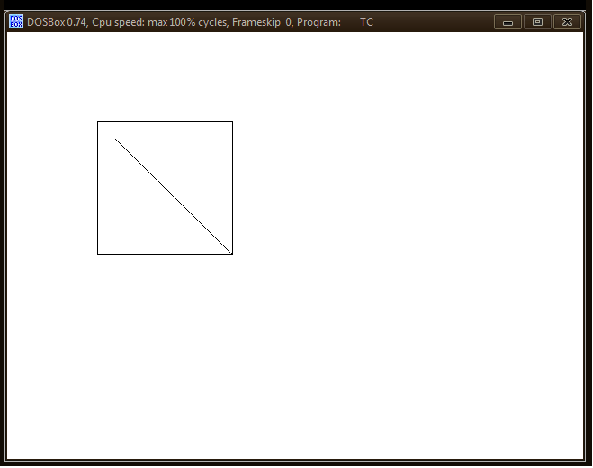
}

***Output :***

Line before clipping :



Line after clipping :



***LINE CLIPPING USING COHEN SUTHERLAND :***

#include<stdio.h>

#include<conio.h>

#include<graphics.h>

void main()

{

int gd=DETECT, gm;

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

float start[4],end[4],code[4];

clrscr();

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

printf("\n\tEnter the bottom-left coordinate of viewport: ");

scanf("%f %f",&xmin,&ymin);

printf("\n\tEnter the top-right coordinate of viewport: ");

scanf("%f %f",&xmax,&ymax);

printf("\nEnter the coordinates for starting point of line: ");

scanf("%f %f",&x1,&y1);

printf("\nEnter the coordinates for ending point of line: ");

scanf("%f %f",&x2,&y2);

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

{

start[i]=0;

end[i]=0;

}

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

if(x1 <xmin) start[0]=1;

if(x1 >xmax) start[1]=1;

if(y1 >ymax) start[2]=1;

if(y1 <ymin) start[3]=1;

if(x2 <xmin) end[0]=1;

if(x2 >xmax) end[1]=1;

if(y2 >ymax) end[2]=1;

if(y2 <ymin) end[3]=1;

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

code[i]=start[i]&&end[i];

if((code[0]==0)&&(code[1]==0)&&(code[2]==0)&&(code[3]==0))

{

if((start[0]==0)&&(start[1]==0)&&(start[2]==0)&&(start[3]==0)&&(end[0]==0)&&(end[1]==0)&&(end[2]==0)&&(end[3]==0))

{

cleardevice();

printf("\n\t\tThe line is totally visible\n\t\tand not a clipping candidate");

rectangle(xmin,ymin,xmax,ymax);

line(x1,y1,x2,y2);

getch();

}

else

{

cleardevice();

printf("\n\t\tLine is partially visible");

rectangle(xmin,ymin,xmax,ymax);

line(x1,y1,x2,y2);

getch();

if((start[2]==0)&&(start[3]==1))

{

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

  y1=ymin;

}

if((end[2]==0)&&(end[3]==1))

{

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

y2=ymin;

}

if((start[2]==1)&&(start[3]==0))

{

  x1=x1+(ymax-y1)/m;

y1=ymax;

}

if((end[2]==1)&&(end[3]==0))

{

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

y2=ymax;

}

if((start[1]==0)&&(start[0]==1))

{

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

x1=xmin;

}

if((end[1]==0)&&(end[0]==1))

{

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

  x2=xmin;

}

if((start[1]==1)&&(start[0]==0))

{

  y1=y1+m\*(xmax-x1);

  x1=xmax;

}

if((end[1]==1)&&(end[0]==0))

{

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

x2=xmax;

}

clrscr();

cleardevice();

printf("\n\t\tAfter clippling:");

rectangle(xmin,ymin,xmax,ymax);

line(x1,y1,x2,y2);

getch();

}

}

else

{

clrscr();

cleardevice();

printf("\nLine is invisible");

rectangle(xmin,ymin,xmax,ymax);

}

getch();

closegraph();

}

***Output :***

