**Practical No: 11**

**Practical Title:** Program to Implement Cohen- Sutherland algorithm for line clipping.

**Roll no:** **Batch: Date of performance:**

#include<stdio.h>

#include<stdlib.h>

#include<math.h>

#include<dos.h>

#include<graphics.h>

#include<conio.h>

typedef struct coordinate

{ int x,y;

char code[4];

}pt;

void drawwindow();

void drawline(pt p1,pt p2,int cl);

pt setcode(pt p);

pt resetendpt(pt p1,pt p2);

int visibility(pt p1,pt p2);

void main()

{ int gd=DETECT,gm,v;

pt p1,p2,ptemp;

initgraph(&gd,&gm,"..\\bgi");

cleardevice();

printf("\n\tEnter end-point 1(x,y)");

scanf("%d %d",&p1.x,&p1.y);

printf("\n\tEnter end-point 2(x,y)");

scanf("%d %d",&p2.x,&p2.y);

cleardevice();

drawwindow();

getch();

drawline(p1,p2,15);

getch();

p1=setcode(p1);

p2=setcode(p2);

v=visibility(p1,p2);

switch(v)

{ case 0:cleardevice();

drawwindow();

drawline(p1,p2,15);

break;

case 1:cleardevice();

drawwindow();

break;

case 2:

cleardevice();

p1=resetendpt(p1,p2);

p2=resetendpt(p2,p1);

drawwindow();

drawline(p1,p2,15);

break; }

getch();

closegraph(); }

void drawwindow()

{ setcolor(RED);

line(150,100,450,100);

line(450,100,450,350);

line(450,350,150,350);

line(150,350,150,100); }

void drawline(pt p1,pt p2,int cl)

{ setcolor(cl);

line(p1.x,p1.y,p2.x,p2.y); }

pt setcode(pt p)

{ pt ptemp;

if(p.y<100)

ptemp.code[0]='1';

else

ptemp.code[0]='0';

if(p.y>300)

ptemp.code[1]='1';

else

ptemp.code[1]='0';

if(p.x>450)

ptemp.code[2]='1';

else

ptemp.code[2]='0';

if(p.x<150)

ptemp.code[3]='1';

else

ptemp.code[3]='0';

ptemp.x=p.x;

ptemp.y=p.y;

return(ptemp); }

int visibility(pt p1,pt p2)

{ int i,flag=0;

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

if(p1.code[i]!='0'||(p2.code[i]!='0'));flag=1;

if(flag==0)

return(1);

else

return(2); }

pt resetendpt(pt p1,pt p2)

{ pt temp;

int x,y,i;

float m,k;

if(p1.code[3]=='1')

x=150;

if(p1.code[2]=='1')

x=450;

if((p1.code[3]=='1')||(p1.code[2]=='1'))

{ m=(float)(p2.y-p1.y)/(p2.x-p1.x);

k=(p1.y+(m\*(x-p1.x)));

temp.y=k;

temp.x=x;

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

temp.code[i]=p1.code[i];

if(temp.y<=350&&temp.y>=100)

return(temp); }

if(p1.code[0]=='1')

y=100;

if(p1.code[1]=='1')

y=350;

if((p1.code[0]=='1')||(p1.code[1]=='1'))

{ m=(float)(p2.y-p1.y)/(p2.x-p1.x)/m;

temp.x=k;

temp.y=y;

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

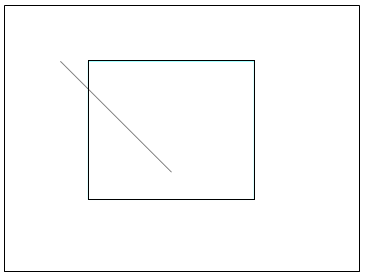
temp.code[i]=p1.code[i];

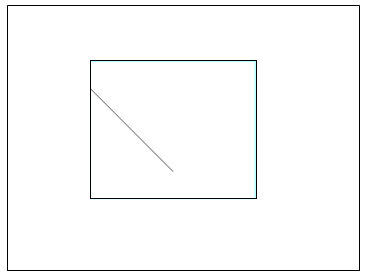
return(temp); }

else

return(p1); }

**OUTPUT: Before Clipping**



****

**After Line Clipping:**