**Practical No: 11**

**Aim :** Write a C Program to perform Cohen-Sutherland line clipping algorithm.

**Performed By :**

**Class :** BCA-III SEM-V

**Date :** 23/09/2024

**Code :**

#include<graphics.h>

#include<conio.h>

#include<stdio.h>

#include<math.h>

int main()

{

int rcode\_begin[4]={0,0,0,0},rcode\_end[4]={0,0,0,0},region\_code[4];

int W\_xmax,W\_ymax,W\_xmin,W\_ymin,flag=0;

float slope;

int x,y,x1,y1,i, xc,yc;

int gr=DETECT,gm;

initgraph(&gr,&gm,"C:\\TURBOC3\\BGI");

printf("\n\*\*\*\*\*\* Cohen Sutherlsnd Line Clipping algorithm \*\*\*\*\*\*\*\*\*\*\*");

printf("\n Now, enter XMin, YMin =");

scanf("%d %d",&W\_xmin,&W\_ymin);

printf("\n First enter XMax, YMax =");

scanf("%d %d",&W\_xmax,&W\_ymax);

printf("\n Please enter intial point x and y= ");

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

printf("\n Now, enter final point x1 and y1= ");

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

cleardevice();

rectangle(W\_xmin,W\_ymin,W\_xmax,W\_ymax);

line(x,y,x1,y1);

line(0,0,600,0);

line(0,0,0,600);

if(y>W\_ymax) {

rcode\_begin[0]=1;

flag=1 ;

}

if(y<W\_ymin) {

rcode\_begin[1]=1;

flag=1;

}

if(x>W\_xmax) {

rcode\_begin[2]=1;

flag=1;

}

if(x<W\_xmin) {

rcode\_begin[3]=1;

flag=1;

}

if(y1>W\_ymax){

rcode\_end[0]=1;

flag=1;

}

if(y1<W\_ymin) {

rcode\_end[1]=1;

flag=1;

}

if(x1>W\_xmax){

rcode\_end[2]=1;

flag=1;

}

if(x1<W\_xmin){

rcode\_end[3]=1;

flag=1;

}

if(flag==0)

{

printf("No need of clipping as it is already in window");

}

flag=1;

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

region\_code[i]= rcode\_begin[i] && rcode\_end[i] ;

if(region\_code[i]==1)

flag=0;

}

if(flag==0)

{

printf("\n Line is completely outside the window");

}

else{

slope=(float)(y1-y)/(x1-x);

if(rcode\_begin[2]==0 && rcode\_begin[3]==1)

{

y=y+(float) (W\_xmin-x)\*slope ;

x=W\_xmin;

}

if(rcode\_begin[2]==1 && rcode\_begin[3]==0)

{

y=y+(float) (W\_xmax-x)\*slope ;

x=W\_xmax;

}

if(rcode\_begin[0]==1 && rcode\_begin[1]==0)

{

x=x+(float) (W\_ymax-y)/slope ;

y=W\_ymax;

}

if(rcode\_begin[0]==0 && rcode\_begin[1]==1)

{

x=x+(float) (W\_ymin-y)/slope ;

y=W\_ymin;

}

if(rcode\_end[2]==0 && rcode\_end[3]==1)

{

y1=y1+(float) (W\_xmin-x1)\*slope ;

x1=W\_xmin;

}

if(rcode\_end[2]==1 && rcode\_end[3]==0)

{

y1=y1+(float) (W\_xmax-x1)\*slope ;

x1=W\_xmax;

}

if(rcode\_end[0]==1 && rcode\_end[1]==0)

{

x1=x1+(float) (W\_ymax-y1)/slope ;

y1=W\_ymax;

}

if(rcode\_end[0]==0 && rcode\_end[1]==1)

{

x1=x1+(float) (W\_ymin-y1)/slope ;

y1=W\_ymin;

}

}

delay(1000);

clearviewport();

rectangle(W\_xmin,W\_ymin,W\_xmax,W\_ymax);

line(0,0,600,0);

line(0,0,0,600);

setcolor(RED);

line(x,y,x1,y1);

getch();

closegraph();

}

**Output :**

