**Assignment No:-5.2.1**

**Assignment Name:-Windowing and Clipping Algorithm.**

**b)Line Clipping Algorithms**

**1)Sutherland & Cohen algorithm**

**Roll No:-12**

**Name:-Chaudhari Priyanka Subhash**

#include<graphics.h>

#include<iostream.h>

#include<stdlib.h>

#include<conio.h>

#include<math.h>

#include<dos.h>

class POINT

{

public:

int x,y;

int bit\_code[4];

};

class GRAPHICS\_12

{

int wx1,wx2,wy1,wy2;

public:

void INIT\_GRAPHICS();

//void READ\_LINE();

void DRAW\_LINE(POINT,POINT);

void DRAW\_WINDOW();

void COHEN\_SUTHERLAND\_CLIPPING();

void SET\_REGION\_CODE(POINT &);

int IS\_VISIBILITY(POINT,POINT);

POINT RESET\_END\_POINTS(POINT,POINT);

};

void GRAPHICS\_12::INIT\_GRAPHICS()

{

/\* request auto detection \*/

int gdriver = DETECT, gmode, errorcode;

/\* initialize graphics and local variables \*/

initgraph(&gdriver, &gmode, "C:\\TURBOC3\\BGI");

}

void GRAPHICS\_12::DRAW\_LINE(POINT p1,POINT p2)

{

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

}

void GRAPHICS\_12::DRAW\_WINDOW()

{

setcolor(1);

rectangle(wx1,wy1,wx2,wy2);

}

void GRAPHICS\_12::SET\_REGION\_CODE(POINT &p)

{

if(p.x<wx1)//left

p.bit\_code[3]=1;

else

p.bit\_code[3]=0;

if(p.x>wx2)//right

p.bit\_code[2]=1;

else

p.bit\_code[2]=0;

if(p.y<wy1)//above

p.bit\_code[0]=1;

else

p.bit\_code[0]=0;

if(p.y>wy2)//below

p.bit\_code[1]=1;

else

p.bit\_code[1]=0;

}

void GRAPHICS\_12::COHEN\_SUTHERLAND\_CLIPPING()

{

int ch;

POINT p1,p2;

cout<<"\n Enter the window pts : ";

cin>>wx1>>wy1>>wx2>>wy2;

cout<<"\n Enter the point p1 : ";

cin>>p1.x>>p1.y;

cout<<"\n Enter the point p2 : ";

cin>>p2.x>>p2.y;

DRAW\_WINDOW();

DRAW\_LINE(p1,p2);

getch();

cleardevice();

cout<<"\n The Clipping window is : ";

DRAW\_WINDOW();

setcolor(6);

SET\_REGION\_CODE(p1);

SET\_REGION\_CODE(p2);

ch=IS\_VISIBILITY(p1,p2);

switch(ch)

{

case 0 : break;

case 1 :

p1=RESET\_END\_POINTS(p1,p2);

p2=RESET\_END\_POINTS(p2,p1);

DRAW\_LINE(p1,p2);

break;

case 2 :

DRAW\_LINE(p1,p2);

break;

}

}

int GRAPHICS\_12::IS\_VISIBILITY(POINT pt1,POINT pt2)

{

for(int i=1;i<=4;i++)

{

if(pt1.bit\_code[i]!=0 || pt2.bit\_code[i]!=0)

break;

}

if(i>=4)

return 2;

//Logican ANDING

for(i=1;i<=4;i++)

{

if(pt1.bit\_code[i]==1 && pt2.bit\_code[i]==1)

{

return 0;

}

}

return 1;

}

POINT GRAPHICS\_12::RESET\_END\_POINTS(POINT pt1,POINT pt2)

{

int x,y,i,ch,dx,dy;

float m,k;

dx=pt2.x-pt1.x;

dy=pt2.y-pt1.y;

//--------finding m

if(dx==0)

dx++;

if(dy==0)

dy++;

if(pt1.bit\_code[3]==1)

x=wx1;

if(pt1.bit\_code[2]==1)

x=wx2;

if((pt1.bit\_code[3]==1)||(pt1.bit\_code[2]==1))

{

m=(float)dy/dx;

k=(float)(pt1.y+(m\*(x-pt1.x)));

pt1.y=k;

pt1.x=x;

if(pt1.y>wy1 && pt1.y<wy2)

return(pt1);

}

if(pt1.bit\_code[0]==1)

y=wy1;

if(pt1.bit\_code[1]==1)

y=wy2;

if((pt1.bit\_code[0]==1)||(pt1.bit\_code[1]==1))

{

m=(float)dy/dx;

k=(float)pt1.x+(float)(y-pt1.y)/m;

pt1.x=k;

pt1.y=y;

return(pt1);

}

else

return(pt1);

}

void main()

{

clrscr();

GRAPHICS\_12 obj;

obj.INIT\_GRAPHICS();

obj.COHEN\_SUTHERLAND\_CLIPPING();

getch();

}