Cohen-Sutherland Algorithm(Line Clipping):

#include<iostream.h>  
#include<conio.h>  
#include<graphics.h>  
static int LEFT=1,RIGHT=2,BOTTOM=4,TOP=8,xl,yl,xh,yh;  
int getcode(int x,int y){  
int code = 0;  
//Peform Bitwise OR to get outcode  
if(yyl) code |=BOTTOM;  
if(xxh) code |=RIGHT;  
return code;  
}  
void main(){  
int gdriver = DETECT,gmode;  
initgraph(&gdriver,&gmode,”C:\TC\BGI”);  
setcolor(BLUE);  
cout<>xl>>yl>>xh>>yh;  
rectangle(xl,yl,xh,yh);  
int x1,y1,x2,y2;  
cout<>x1>>y1>>x2>>y2;  
line(x1,y1,x2,y2);  
getch();

int outcode1=getcode(x1,y1), outcode2=getcode(x2,y2);  
int accept = 0;                                                                                              //decides if line is to be drawn  
while(1){  
float m =(float)(y2-y1)/(x2-x1);  
if(outcode1==0&&outcode2==0){                                                     //Both points inside. Accept line  
accept = 1;  
break;  
}else if((outcode1 & outcode2)!=0){                                                  //AND of both codes != 0.Line is outside. Reject line  
break;  
}else{  
int x,y;  
int temp;  
if(outcode1==0) temp = outcode2;                                                   //Decide if point1 is inside. if not calculate intersection  
else temp = outcode1;

if(temp & TOP){                                   //Line clips top edge  
x = x1+ (yh-y1)/m;  
y = yh;  
}else if(temp & BOTTOM){              //Line clips bottom edge  
x = x1+ (yl-y1)/m;  
y = yl;  
}else if(temp & LEFT){                    //Line clips left edge  
x = xl;  
y = y1+ m\*(xl-x1);  
}else if(temp & RIGHT){                //Line clips right edge  
x = xh;  
y = y1+ m\*(xh-x1);  
}  
if(temp == outcode1){                   //Check which point we had selected earlier as temp, and replace its co-ordinates  
x1 = x;  
y1 = y;  
outcode1 = getcode(x1,y1);  
}else{  
x2 = x;  
y2 = y;  
outcode2 = getcode(x2,y2);  
}  
}  
}  
setcolor(WHITE);  
cout<<“After clipping:”;  
if(accept) line(x1,y1,x2,y2);  
getch();  
closegraph();  
}