#include<iostream> #include<graphics.h> using namespace std;

static int LEFT=1,RIGHT=2,BOTTOM=4,TOP=8,xl,yl,xh,yh;

int getcode(int x,int y){

int code = 0;

//Perform Bitwise OR to get outcode if(y > yh) code |=TOP; if(y < yl) code |=BOTTOM; if(x < xl) code |=LEFT; if(x > xh) code |=RIGHT; return code;

}

int main()

{

int gdriver = DETECT,gmode; initgraph(&gdriver,&gmode,NULL);

setcolor(BLUE);

cout<<"Enter bottom left and top right co-ordinates of window: "; cin>>xl>>yl>>xh>>yh; rectangle(xl,yl,xh,yh); int x1,y1,x2,y2;

cout<<"Enter the endpoints of the line: "; cin>>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); //Both points inside. Accept line if(outcode1==0 && outcode2==0){

accept = 1; break;

}

//AND of both codes != 0.Line is outside. Reject line

else if((outcode1 & outcode2)!=0){

break; }else{ int x,y;

int temp;

//Decide if point1 is inside, if not, calculate intersection

if(outcode1==0) temp = outcode2; else

temp = outcode1;

//Line clips top edge if(temp & TOP){ x = x1+ (yh-y1)/m;

y = yh;

}

else if(temp & BOTTOM){ //Line clips bottom edge

1. = x1+ (yl-y1)/m;
2. = yl;

}else if(temp & LEFT){ //Line clips left edge

1. = xl;
2. = y1+ m\*(xl-x1);

}else if(temp & RIGHT){ //Line clips right edge

1. = xh;
2. = y1+ m\*(xh-x1);

}

//Check which point we had selected earlier as temp, and replace its coordinates

if(temp == outcode1){

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); return 0;

closegraph();

}