

/*

Title: - Write C++ program to implement Cohen Southerland line clipping algorithm.

Roll No:-

Class:-SE Computer

Sub:-OOPL & CGL

Date:-

*****/

Program-

```
#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
```

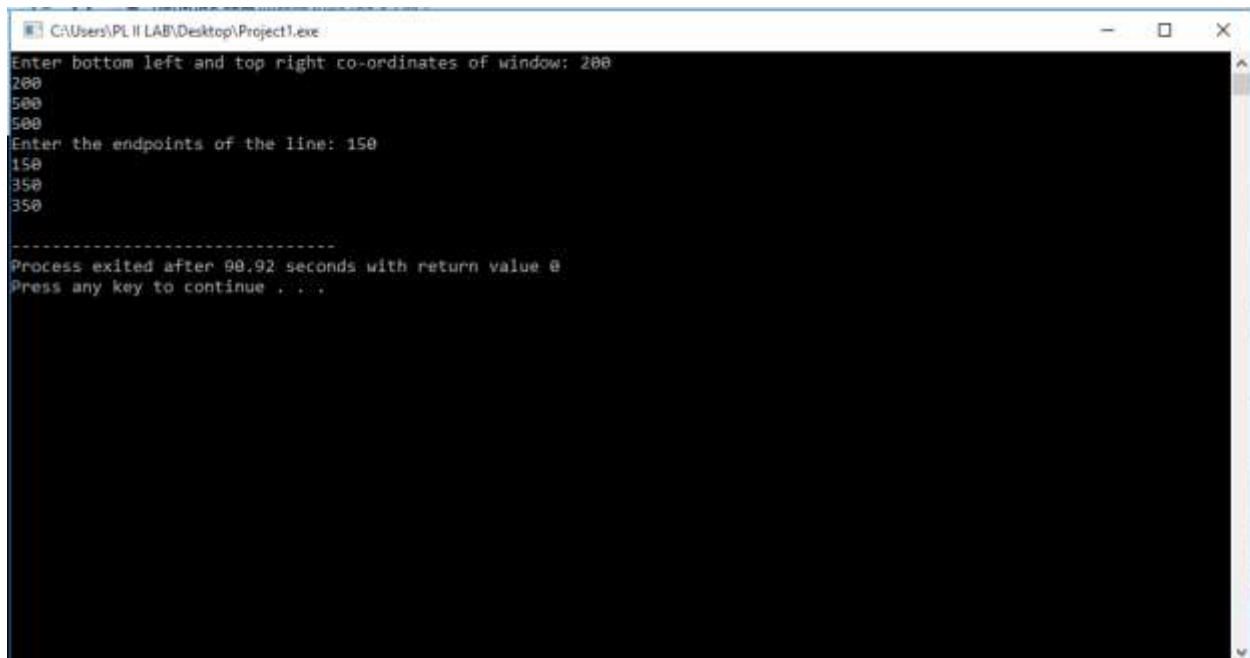
```

        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
                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);
            }
            //Check which point we had selected earlier as temp, and replace its co-
ordinates
            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)

```

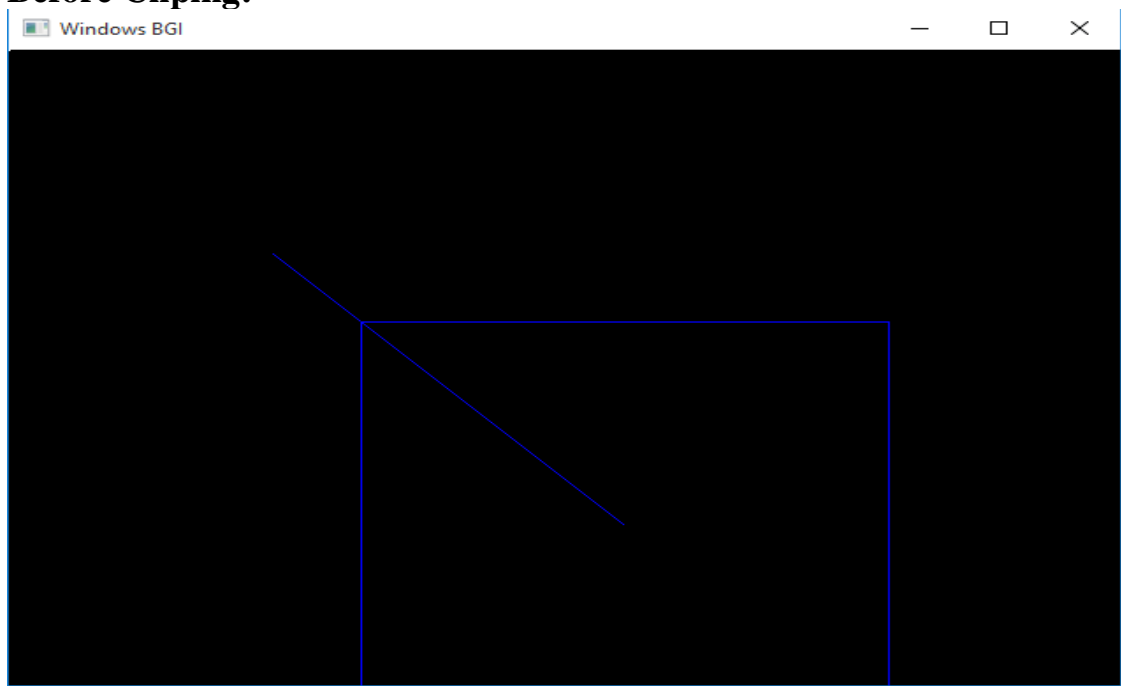
```
        line(x1,y1,x2,y2);  
    return 0;  
    closegraph();  
}
```

/*Output:-



```
C:\Users\PLI LAB\Desktop\Project1.exe  
Enter bottom left and top right co-ordinates of window: 200  
500  
500  
Enter the endpoints of the line: 150  
350  
350  
-----  
Process exited after 90.92 seconds with return value 0  
Press any key to continue . . .
```

Before Clipping:



After Clipping:-

