Name:- Ayush Godiyal Course:- BCA Sem:- 6 Sec:- A
University Roll No:- 1121033 Subject:- CG practical Exam.

Answer ②: Cohen Sutherland Line Clipping

Algorithm

step 1:- Start

step 2:- Read two end points of line as $p1(x_1, y1)$ and $p2(x_2, y2)$

step 3:- Read two corner points of the clipping window (left-top and right-bottom) as $(wx1, wy1)$ and $(wx2, wy2)$

step 4:- Assign the regions codes for 2 endpoints P1 and P2 using following steps:-

initialize code with 0000

Set bit 1 if $x < wx1$

set bit 2 if $x > wx2$

set bit 3 if $y < wy2$

set bit 4 if $y > wy1$

step 5:- check for visibility of line

(a.) If region codes for both endpoints are zero then line is completely visible. Draw the line go to step 10

(b.) If region codes for endpoints are not zero and logical ANDing of them is also non zero then line is invisible. Discard the line and move to step 10

(c) If it does not satisfy 5.a and 5.b then line is partially visible.

(Step 6 :- Determine the intersecting edge of clipping windows as follows :-

(a) If region codes for both endpoints are non-zero find intersection points p1' and p2' with boundary edges.

(b) If region codes for any one end point is non zero then find intersection point p1' or p2'

step 7 :- Divide the line segment considering intersection points.

step8:- Reject line segment if any end point of line appears outside of any boundary.

step 9:- Draw the clipped line segment.

step10:- stop!

Program

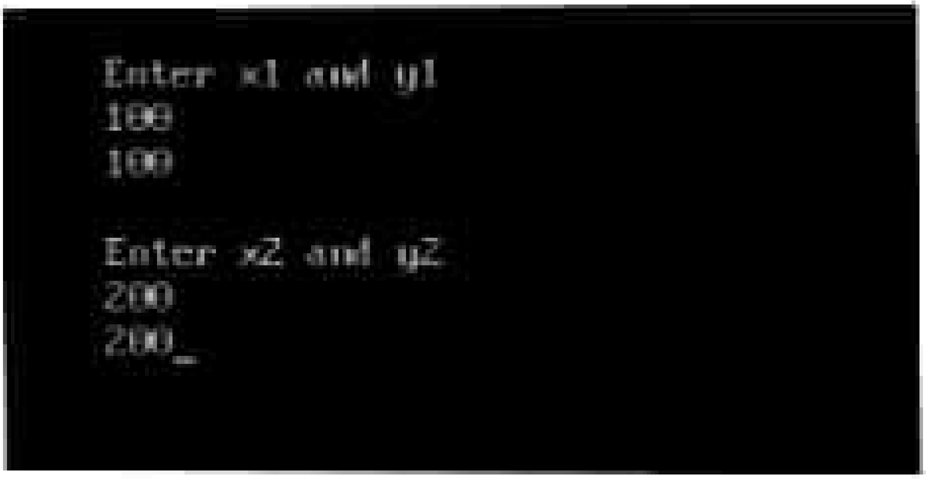
```
#include <stdio.h>
#include <stdlib.h>
#include <math.h>
#include <graphics.h>
#include <dos.h>

typedef struct coordinate
{
int x, y;
char code[4];
} PT;
void drawwindow();
void drawline(PT, P1, PT, P2);
PT setcode(PT, P);
int visibility(PT P1, PT P2);
PT resetendpt(PT P1, PT, P2);
void main()
{
```

```
int gd = DETECT , V, gm;
PT p1 , p2 , p3 , p4 , ptemp ;
printf ("\n Enter x1 and y1\n");
scanf ("%d %d", &p1.x , &p1.y);
printf ("\n Enter x2 and y2 \n");
scanf (" %d %d", &p2.x , &p2.Y);
initgraph ( &gd , &gm , " ");
drawwindow ();
delay (500);
drawline (P1, P2);
delay (500);
cleardevice ();
delay (500);
P1 = setcode (p1);
p2 = setcode (p2);
v = visiblity (P1, P2);
delay (500);
switch (v)
{
case 0 : drawwindow ();
 delay (500);
 drawline (P1, P2);
 Break;
 Case 1: drawwindow( );
 delay (500);
break ;
Case 2: p3 = resetendpt (P1, P2);
p4 = resetendpt (P2, P1);
drawwindow ();
delay (500);
drawline (P3, P4);
break;
}
delay (5000);
closegraph ();
}
void drawwindow ()
{
line (150, 100, 450, 100);
line (450, 100, 450, 350);
line (450, 350, 150, 350);
~~line~~
line (150, 350, 150, 100);
}
Void drawline ( PT P1 , PT P2)
{
line ( P1.x , P1.y , P2.x , P2.y );
}
PT setcode ( PT p)
{
PT ptemp;
if ( p.y<100)
ptemp.code[0]='1';
else
ptemp.code [0] = '0';
```

```
IF (p.y>350)
Ptemp.code[1] = '1';
else
ptemp.code [1]='0';

IF (p.x >450)
ptemp.code[2] = '1';
else
ptemp.code [2] = '0';

IF ( p.x <150)
Ptemp.code(3] = '1';
else
ptemp.code[3] = '0';

ptemp.x = p.x;
Ptemp.y = p.y;
return (ptemp);
}
int visibility (PT p1, PT p2)
{
int i, flag=0;
for ( i=0; i<4; i++ )
{
if( (p1.code[i]!='0') || (p2.code[i]!='0'))
flag = 1;
}
if (flag==0)
return (0);
for (i=0; i<4; i++)
{
```

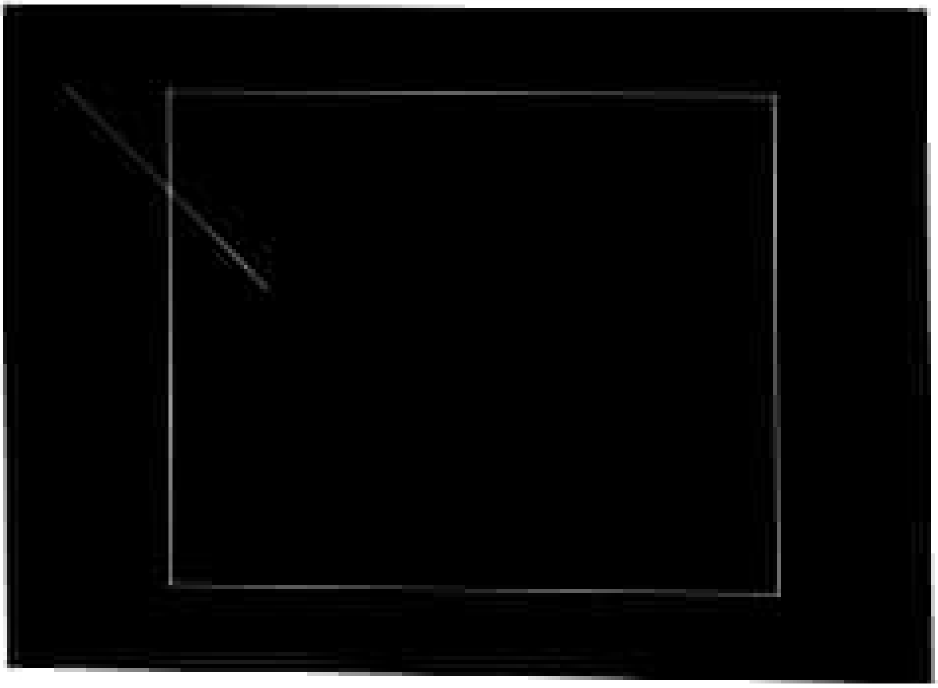
```
{
if (p1.code [i] == p2.code[i]) && (p1.code[i] == '1'))
flag = '0';
}
if (flag == 0)
return (1);
return (2);
}
PT resentendpt (PT p1, PT p2)
{
PT temp;
int x, y, i;
float m, k;
if (p1.code [3] == '1')
x = 150;
if (p1.code [2] == '1')
x = 450;
if ((p1.code [3] == '1') || (p1.code [2] == '1'))
{
m = (float)(p2.y - p1.y) / (p2.x - p1.x);
k = (p1.y + (m * (x - p1.x)));
temp.y = k;
temp.x = x;
for (i = 0; i<4; i++)
temp.code [i] = p1.code [i];
if (temp.y <= 350 && temp.y >= 100)
return (temp);
}
```

```
if (P1.code[0] =='1')
y=100;
if (P1 .code[1] =='1')
y=380;
if((p1.code[0]=='1')||(P1.code[1]=='1'))
{
m=(float)(P2.y-P1.y)/(P2. x-P1.x);
k= (float)(P1.x +(float)(y-P1.y)/m;
temp.x =k;
temp.y=y;
for(i=0; i<4; i++)
temp.code[i]=p1.code[i];
return(temp);
}
else
return(P1);
}
```



```
Enter x1 and y1
100
100

Enter x2 and y2
200
200_
```

Before Clipping



After Clipping