Program 5

Aim: Write a program to clip line using Cohen Sutherland Algorithm.

Theory:

The **Cohen–Sutherland algorithm** is a computer-graphics algorithm used for line clipping. The algorithm divides a two-dimensional space into 9 regions and then efficiently determines the lines and portions of lines that are visible in the central region of interest (the viewport).

The algorithm was developed in 1967 during flight-simulator work by Danny Cohen and Ivan Sutherland.

Algorithm:

NEW

1. Calculate positions of both endpoints of the line.

2. Perform OR operation on both of these end-points.

3. If the OR operation gives 0000
   Then
   line is considered to be visible
   else
   Perform AND operation on both endpoints
   If And ≠ 0000
   then the line is invisible
   else
   And=0000
   Line is considered the clipped case.

4. If a line is clipped case, find an intersection with boundaries of the window
   $m=(y_2-y_1)(x_2-x_1)$

(a) If bit 1 is "1" line intersects with left boundary of rectangle window
$y_3=y_1+m(x-X_1)$
where $X = X_{wmin}$
where $X_{wmin}$is the minimum value of X co-ordinate of window

(b) If bit 2 is "1" line intersect with right boundary
$y_3=y_1+m(X-X_1)$
where $X = X_{wmax}$
where X more is maximum value of X co-ordinate of the window

(c) If bit 3 is "1" line intersects with bottom boundary

$X_3=X_1+(y-y_1)/m$

where $y = y_{wmin}$

y_{wmin} is the minimum value of Y co-ordinate of the window

(d) If bit 4 is "1" line intersects with the top boundary

$X_{3=X}1+(y-y_1)/m$

where $y = y_{wmax}$

y_{wmax} is the maximum value of Y co-ordinate of the window

Code:

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

typedef struct coordinates{
    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);

int main(){
    int v;
    PT p1,p2,p3,p4,ptemp;
    cout<<"\nEnter x1 and y1\n";
    cin>>p1.x>>p1.y;
    cout<<"\nEnter x2 and y2\n";
    cin>>p2.x>>p2.y;

    int g_mode, g_driver = DETECT;
    initgraph(&g_driver, &g_mode, "C:\\TURBOC3\\BGI");

    drawwindow();
```

```
    delay(500);

    drawline(p1,p2);
    delay(500);
    cleardevice();

    delay(500);
    p1=setcode(p1);
    p2=setcode(p2);
    v=visibility(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();
    return 0;
}

void drawwindow(){
    line(150,100,450,100);
    line(450,100,450,350);
    line(450,350,150,350);
    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 resetendpt(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=350;

    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);
}
```

Output:

```
Enter x1 and y1
100
50

Enter x2 and y2
1000
900_
```