AISSMS
COLLEGE OF ENGINEERING
Approved by AICTE, New Delhi, Recognized by
Govt. of Maharashtra, Affiliated to Savitribai Phule Pune University
and recognized 2(f) and 12(B) by UGC (Id.No. PU / PN/ Engg. / 093 (1992)
Accredited by NAAC with 'A+' Grade

*Name :- Pratik Pingale* *Class :- SE Comp. 1*

*Roll no. :- 19CO056*

# Experiment A-6

## Aim:

Write C++ program to implement Cohen Southerland line clipping algorithm

## Algorithm

**Step 1**: Read (Xmin,Ymin) and (Xmax, Ymax) - Lower-left and top-right corner of the clipping window.

**Step 2**: Read A(x1. y1) and B(x, y2) end points of line.

**Step 3**: Compute outcode of A and B

if y1 > Ymax then Outcode A(1) = 1 else 0 if y2 > Ymax then Outcode_B(1) = 1 else 0

if y1 < Ymin then Outcode_A(2) = 1 else 0 if y2 < Ymin then OutCode B/2) = 1 else 0

if x1 > Xmax then Outcode_A(3) =1 else 0 if x2 > Xmax then Outcode B(3) = 1 else 0

if x1 < Xmin then Outcode_A(4) = 1 else 0 if x2 < Xmin then Outcode B(4)=1 else 0

**Step 4**: If Outcode_A OR Outcode_B = = 0000 then

Display entire line and goto step 5

else if Outcode A AND Outcode B +0000 then

Reject the entire line

else

Compute the intersection point with window boundaries

Repent Step 4.

**Step 5**: Stop

## Program :-

```
#include<iostream>
#include<graphics.h>

using namespace std;

class lineClip {
    private:
        int RIGHT = 2, LEFT = 1, TOP = 8, BOTTOM = 4;
        int x1, x2, y1, y2, xl, yl, xh, yh, x, y;

    public:
        int getcode(int x, int y);
        void process();
};

int lineClip::getcode(int x, int y) {
    int code = 0;
    if (y > yh)
        code |= TOP;
    if (y < yl)
        code |= BOTTOM;
    if (x < xl)
        code |= LEFT;
    if (x > xh)
        code |= RIGHT;
    return code;
}

void lineClip::process() {
    int code1, code2;

    cout << "\n Enter the botttom left and upper right coordinate of the
rectangle : ";
    cin >> xl >> yl >> xh >> yh;

    setcolor(YELLOW);
    rectangle(xl, yl, xh, yh);

    cout << "\n Enter the line coordinate :";
    cout << "\n Starting Coordinate :";
    cin >> x1 >> y1;
    cout << "\n Ending coordinate : ";
    cin >> x2 >> y2;

    setcolor(WHITE);
    line(x1, y1, x2, y2);
    delay(1000);

    code1 = getcode(x1, y1);
```

```
    code2 = getcode(x2, y2);
    int temp;

    float m;
    int flag = 0;
    while (1) {
        m = (float)(y2 - y1) / (x2 - x1);

        if (code1 == 0 && code2 == 0) {
            flag = 1;
            break;
        } else if ((code1 & code2) != 0) {
            break;
        } else {
            if (code1 == 0)
                temp = code2;
            else
                temp = code1;

            if (temp & TOP) {
                x = x1 + (yh - y1) / m;
                y = yh;
            } else if (temp & BOTTOM) {
                x = x1 + (yl - y1) / m;
                y = yl;
            } else if (temp & LEFT) {
                y = y1 + m * (xl - x1);
                x = xl;
            } else if (temp & RIGHT) {
                y = y1 + m * (xh - x1);
                x = xh;
            }
            if (temp == code1) {
                x1 = x;
                y1 = y;
                code1 = getcode(x1, y2);
            } else {
                x2 = x;
                y2 = y;
                code2 = getcode(x2, y2);
            }
        }
    }
    cleardevice();
    rectangle(xl, yl, xh, yh);
    setcolor(YELLOW);

    if (flag == 1)
        line(x1, y1, x2, y2);
    getch();
    closegraph();
}
```

```
int main() {
    int gd = DETECT, gm;
    initgraph( & gd, & gm, NULL);
    lineClip l1;
    l1.process();
    return 0;
}
```

# OUTPUT :-

```
proxima@proxzima:~/Documents/Extras/CG/Practicals

~/Doc/E/C/Practicals
 g++ -g "/home/proxima/Documents/Extras/CG/Practicals/clipping.cpp" -o "/home/proxima/Documents/Extras/CG/Practicals/clipping" -lgraph -lGL -lGLU -lglut && "/home/proxima/Documents/Extras/CG/Practicals/clipping"

 Enter the botttom left and upper right coordinate of the rectangle : [xcb] Unknown sequence number while processing queue
[xcb] Most likely this is a multi-threaded client and XInitThreads has not been called
[xcb] Aborting, sorry about that.
clipping: ../../src/xcb_io.c:260: poll_for_event: Assertion `!xcb_xlib_threads_sequence_lost' failed.
0 0 0 0

 Enter the line coordinate :
 Starting Coordinate :10 50

 Ending coordinate : 400 300

~/Doc/E/C/Practicals
```

SDL-libgraph -- Graphics on GNU/Linux