Experiment 7

Student Name: Gaganjot Singh **UID: 22BCS14843**
Branch: B.E. CSE III Yr **Section:** 22BCS-IOT-612-B
Semester: 6th
Subject Name: Computer Graphics with Lab **Subject Code:** 22CSH-352

1. **Aim:** Evaluate 4 bit-region codes for line end points and determine whether the line lies inside or outside the screen.

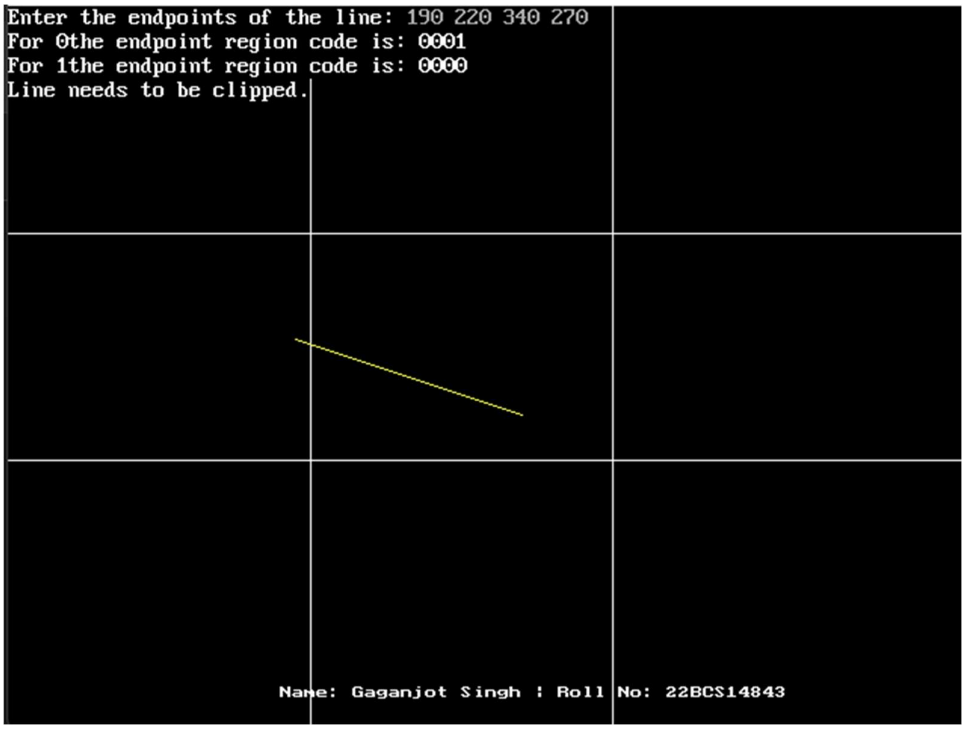
2. **Objective:** To implement and evaluate 4-bit region code for line endpoints and determine whether the line lies inside or outside the screen.

3. **Code:**

```
#include <iostream.h>
#include <graphics.h>
#include <conio.h>
void main()

{
    int gd = DETECT, gm;

    initgraph(&gd, &gm, "C:\\TURBOC3\\BGI");
    outtextxy(180, 450, "Name: Gaganjot Singh | Roll No: 22BCS14843");

    int xmax = 400, ymax = 300, xmin = 200, ymin = 150;
    line(xmin, 0, xmin, getmaxy());
    line(xmax, 0, xmax, getmaxy());
    line(0, ymax, getmaxx(), ymax);
    line(0, ymin, getmaxx(), ymin);
    cout << "Enter the endpoints of the line: ";
    int x[2], y[2], num[2];
    cin >> x[0] >> y[0] >> x[1] >> y[1];
    setcolor(YELLOW);
    line(x[0], y[0], x[1], y[1]);
    for (int i = 0; i < 2; i++)
    {
        int bit1 = 0, bit2 = 0, bit3 = 0, bit4 = 0;
        if (y[i] < ymin)
            bit1 = 1;
        if (y[i] > ymax)
            bit2 = 1;
        if (x[i] > xmax)
            bit3 = 1;
        if (x[i] < xmin)
            bit4 = 1;
        cout << "For " << i << "the endpoint region code is: "
             << bit1 << bit2 << bit3 << bit4 << endl;
        num[i] = bit4 * 1 + bit3 * 2 + bit2 * 4 + bit1 * 8;
    }
    if (!(num[0] | num[1]))
```

```
{
    cout << "Line is completely inside the window." << endl;
}
else if (!(num[0] & num[1]))
{
    cout << "Line needs to be clipped." << endl;
}
else
{
    cout << "Line is completely outside the window." << endl;
}
getch();
closegraph();}
```

4. Output:



5. Learning Outcome:

i. Learned how to represent a point's position relative to a clipping window using a 4-bit region code.

ii. Used bitwise OR (|) to check if a line is inside and bitwise AND (&) to determine if it is completely outside the clipping region.

iii. Implemented Turbo C++ graphics functions (initgraph(), line(), setcolor(), closegraph()) to visualize the clipping window and the input line.

iv. Understood how to classify and determine whether a line is fully visible, completely outside, or requires clipping using the Cohen-Sutherland algorithm principles.