Experiment 6

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:** Analyze and implement the reflection of a point about a line defined by the equation y=mx+c.

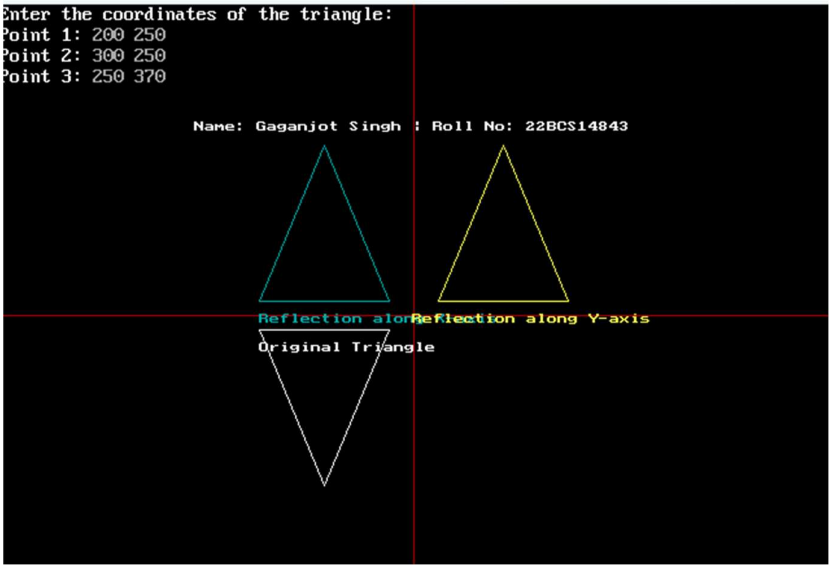
2. **Objective:** To implement and analyze the reflection of a point about a straight line defined by the equation y=mc+c.

3. **Code:**

```
#include <iostream.h>
#include <conio.h>
#include <graphics.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. Learn how to initialize and use the graphics mode in C++ using initgraph(), line(), and outtextxy() functions from the graphics.h library.

ii. Gain practical knowledge of reflection transformations by reflecting a triangle along the X-axis and Y-axis using coordinate manipulation.

iii. Learn how to use getmaxx() and getmaxy() to determine screen boundaries and implement transformations accordingly.

iv. Understand how to take user input for geometric coordinates using cin and display messages using cout for interactive graphics applications.

v. Develop an intuitive understanding of how mathematical transformations (such as reflections) can be visually represented using programming.