Course Title: *Computer Graphics Laboratory*

Course code: *CSE-304*

3rd year 1st semester

Lab Report No: *07*

Submitted to-

Dr. Mohammad Shorif Uddin

Professor

and

Dr. Morium Akter

Associate Professor
Department of Computer Science and Engineering

Jahangirnagar University

Savar, Dhaka-1342

Sl	Class Roll	Exam Roll	Name
01	388	202200	Md.Tanvir Hossain Saon

Experiment Name: Line clipping using Bezier Barsky polynomial approximations.

Introduction: In this lab report, we explore how the Barsky-Bezier algorithm works. We'll see how it tweaks Cohen-Sutherland for curves and uses clever math to reshape curves. A hands-on demo using the graphics.h library will illustrate how we can practically draw and clip Bézier curves, opening a window to the world of graphic manipulation.Through this report, we aim to introduce you to the art of Bézier curve clipping, the magic of Barsky-Bezier, and how these concepts can jazz up your visual creations.

Source Code:

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

// Define a structure to represent a 2D point
struct Point {
    int x, y;
};

// Function to calculate the Bézier curve point using polynomial approximation
Point calculateBezierPoint(Point p[], double t) {
    double u = 1.0 - t;
    double tt = t * t;
    double uu = u * u;
    double uuu = uu * u;
    double ttt = tt * t;

    Point pFinal;
    pFinal.x = static_cast<int>(p[0].x * uuu + 3 * p[1].x * t * uu + 3 * p[2].x * tt * u + p[3].x * ttt);
    pFinal.y = static_cast<int>(p[0].y * uuu + 3 * p[1].y * t * uu + 3 * p[2].y * tt * u + p[3].y * ttt);

    return pFinal;
}
```

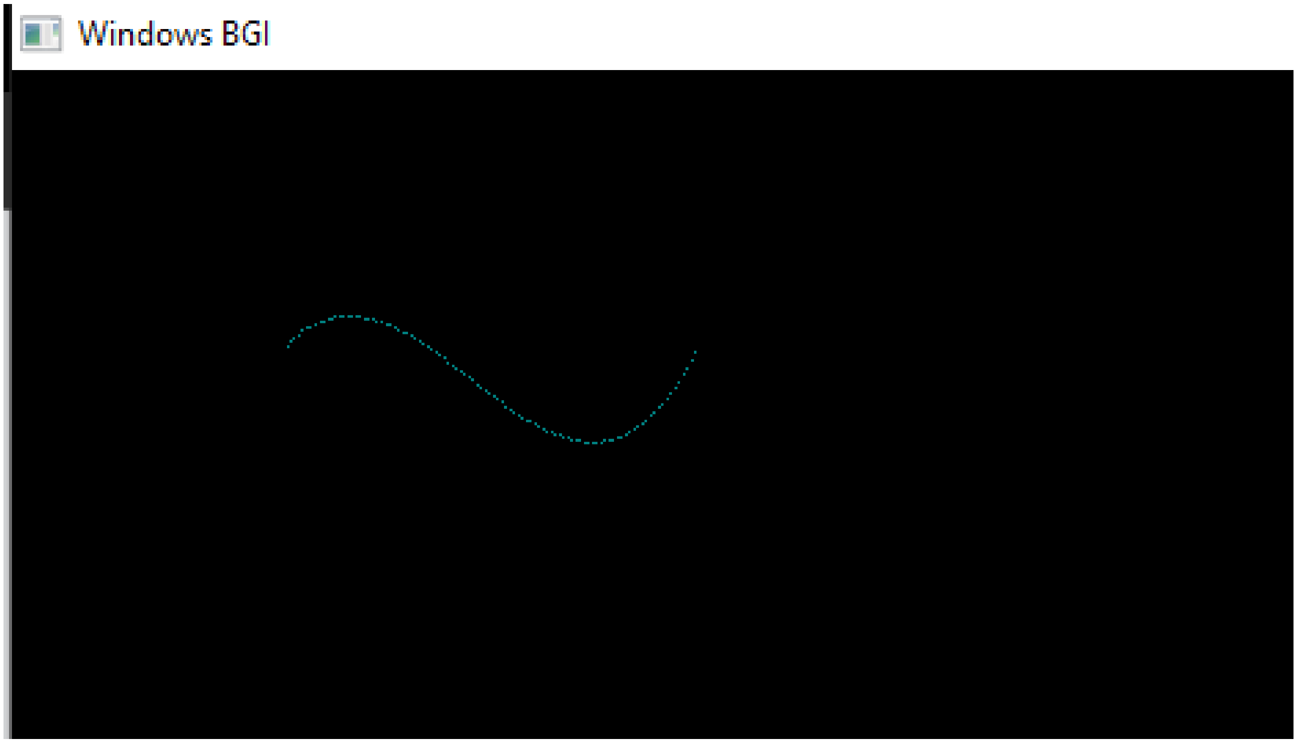
```
// Main function
int main() {
    int gd = DETECT, gm;
    initgraph(&gd, &gm, "C:\\Turboc3\\BGI");

    // Define Bézier control points
    Point controlPoints[4] = {{100, 100}, {150, 50}, {200, 200}, {250, 100}};

    // Draw the Bézier curve using polynomial approximation
    for (double t = 0.0; t <= 1.0; t += 0.01) {
        Point p = calculateBezierPoint(controlPoints, t);
        putpixel(p.x, p.y, WHITE);
    }

    delay(50000); // Pause for a few seconds before closing the graphics window
    closegraph();
    return 0;
}
```

Output:



Discussion:

The Barsky-Bezier algorithm cleverly clips Bézier curves. It adapts Cohen-Sutherland for curves, preserving their shape while trimming. By finding curve-clip intersections, it uses polynomial equations to approximate clipped parts.