**Curve Drawing:**

// A simple program to draw a curve using glut library

#include <GL/gl.h>

#include <GL/glut.h>

#include <math.h>

// Define the number of points to draw the curve

#define N 100

// Define the control points for the curve

float ctrlPoints[4][3] = {

    { -4.0, -4.0, 0.0},

    { -2.0, 4.0, 0.0},

    {2.0, -4.0, 0.0},

    {4.0, 4.0, 0.0}

};

// Define the callback function to display the curve

void display(void)

{

    // Clear the color buffer

    glClear(GL\_COLOR\_BUFFER\_BIT);

    // Set the color to red

    glColor3f(1.0, 0.0, 0.0);

    // Begin drawing the curve as a line strip

    glBegin(GL\_LINE\_STRIP);

    // Loop through the points and evaluate the curve

    for (int i = 0; i <= N; i++)

    {

        // Calculate the parameter t

        float t = (float)i / (float)N;

        // Calculate the point on the curve using the Bernstein polynomials

        float x = (1-t)\*(1-t)\*(1-t)\*ctrlPoints[0][0] + 3\*t\*(1-t)\*(1-t)\*ctrlPoints[1][0] + 3\*t\*t\*(1-t)\*ctrlPoints[2][0] + t\*t\*t\*ctrlPoints[3][0];

        float y = (1-t)\*(1-t)\*(1-t)\*ctrlPoints[0][1] + 3\*t\*(1-t)\*(1-t)\*ctrlPoints[1][1] + 3\*t\*t\*(1-t)\*ctrlPoints[2][1] + t\*t\*t\*ctrlPoints[3][1];

        float z = (1-t)\*(1-t)\*(1-t)\*ctrlPoints[0][2] + 3\*t\*(1-t)\*(1-t)\*ctrlPoints[1][2] + 3\*t\*t\*(1-t)\*ctrlPoints[2][2] + t\*t\*t\*ctrlPoints[3][2];

        // Draw the point on the curve

        glVertex3f(x, y, z);

    }

    // End drawing the curve

    glEnd();

    // Flush the output

    glFlush();

}

// Define the main function

int main(int argc, char \*\*argv)

{

    // Initialize the glut library

    glutInit(&argc, argv);

    // Set the display mode to single buffer and RGB color

    glutInitDisplayMode(GLUT\_SINGLE | GLUT\_RGB);

    // Set the window size and position

    glutInitWindowSize(500, 500);

    glutInitWindowPosition(100, 100);

    // Create the window with the title "Curve Drawing"

    glutCreateWindow("Curve Drawing");

    // Set the display callback function

    glutDisplayFunc(display);

    // Set the background color to white

    glClearColor(1.0, 1.0, 1.0, 0.0);

    // Set the viewing transformation

    //glMatrixMode(GL\_PROJECTION);

    glLoadIdentity();

    gluOrtho2D(-5.0, 5.0, -5.0, 5.0);

    // Enter the main loop

    glutMainLoop();

    // Return success

    return 0;

}