**/\*EXPERIMENT NO.- 7A**

**PROBLEM STATEMENT-** Generate fractal patterns using Bezier curve

**ROLL NO.-** ITSA01

**NAME-** Rahul Devidas Thakare

\*/

**INPUT**

#include<stdio.h>

#include<GL/gl.h>

#include<GL/glu.h>

#include<GL/glut.h>

*void* init(){

    glClearColor(1,0,0,0);

    glColor3f(0,0,1);

    gluOrtho2D(0,640,0,480);

}

*void* draw(){

    glClear(GL\_COLOR\_BUFFER\_BIT);

*int* x[4],y[4],px,py,i;

    for(i=0;i<4;i++){

        x[0]=320;

        x[1]=400;

        x[2]=480;

        x[3]=560;

        x[0]=240;

        x[1]=100;

        x[2]=380;

        x[3]=240;

        glBegin(GL\_LINES);

            glVertex2i(320,0);

            glVertex2i(320,480);

            glVertex2i(0,240);

            glVertex2i(640,240);

        glEnd();

    }

*double* t;

    for(t=0.0;t<=1;t+=0.001){

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

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

        glBegin(GL\_POINTS);

            glVertex2i(px,py);

        glEnd();

    }

    glFlush();

}

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

    glutInit(&argc,argv);

    glutInitDisplayMode(GLUT\_SINGLE | GLUT\_RGB);

    glutInitWindowPosition(0,0);

    glutInitWindowSize(640,480);

    glutCreateWindow("Beizer Curve");

    init();

    glutDisplayFunc(draw);

    glutMainLoop();

    return 0;

}