Name – Isha Kishor Katariya

Class - SEIT Div - A

Roll No – 207A056

Experiment - Generate fractal patterns using i) Bezier ii) Koch Curve

#include <iostream> #include <math.h>

#include <time.h> #include <GL/glut.h> using namespace std;

double x,y,len,angle;

int it; void init(){ glClearColor(1.0,1.0,1.0,0.0)

;

glMatrixMode(GL\_PROJECTION);

gluOrtho2D(0,640,0,480);

glClear(GL\_COLOR\_BUFFER\_BIT);

}

void line1(int x1, int y11, int x2,int y2){ glColor3f(0,1,0); glBegin(GL\_LINES)

; glVertex2i(x1,y11); glVertex2i(x2,y2); glEnd();

glFlush(); }

void k\_curve(double x, double y, double len, double angle, int it){ if(it>0){ len /=3;

k\_curve(x,y,len,angle,(it-1)); x += (len \*

cosl(angle \* (M\_PI)/180)); y += (len \* sinl(angle \* (M\_PI)/180)); k\_curve(x,y, len, angle+60,(it1)); x += (len \* cosl((angle + 60) \* (M\_PI)/180)); y += (len \* sinl((angle +

60) \* (M\_PI)/180)); k\_curve(x,y, len, angle-

60,(it-1)); x += (len \* cosl((angle - 60) \*

(M\_PI)/180)); y += (len \* sinl((angle - 60) \*

(M\_PI)/180)); k\_curve(x,y,len,angle,(it-1));

} else{

line1(x,y,(int)(x + len \* cosl(angle \* (M\_PI)/180) + 0.5),(int)(y + len \*

sinl(angle \* (M\_PI)/180) + 0.5));

} } void Algorithm(){ k\_curve(x,y,len,an

gle,it);

} int main(int argc, char\*\* argv){ cout<<"\n Enter Starting Point x space y "; cin>>x>>y; cout <<"\n Lenght of lineÂ and space angle of line"; cin>>len>>angle; cout<<"\n No. of ittration "; cin>>it; glutInit(&argc, argv);

glutInitDisplayMode(GLUT\_SINGLE|GLUT\_RGB); glutInitWindowSize(640,480); glutInitWindowPosition(200,200); glutCreateWindow("Koch");

init();

glutDisplayFunc(Algorithm); glutMainLoop(); return 0;

}