**UCS 1712 – GRAPHICS AND MULTIMEDIA LAB**

**ASSIGNMENT – 2**

**VISHAL N**

**185001198**

**03.08.2021 CSEC**

**1. DDA:**

#include<GL/glut.h>

#include<math.h>

void myInit() {

    glClearColor(1.0, 1.0, 1.0, 0.0);

    glColor3f(0.0f, 0.0f, 0.0f);

    glPointSize(0.05);

    glMatrixMode(GL\_PROJECTION);

    glLoadIdentity();

    gluOrtho2D(-600.0, 600.0, -600.0, 600.0);

}

void myDDA() {

    glClear(GL\_COLOR\_BUFFER\_BIT);

    glBegin(GL\_LINES);

    glVertex2d(-600, 0);

    glVertex2d(600, 0);

    glEnd();

    glBegin(GL\_LINES);

    glVertex2d(0, -600);

    glVertex2d(0, 600);

    glEnd();

    glBegin(GL\_POINTS);

    glVertex2d(0, 0);

    int X0 = 100, Y0 = 0, X1 = 0, Y1 = 200;

    int dx = X1 - X0;

    int dy = Y1 - Y0;

    int steps = abs(dx) > abs(dy) ? abs(dx) : abs(dy);

    float x\_increment = dx / (float)steps;

    float y\_increment = dy / (float)steps;

    float X = X0;

    float Y = Y0;

    for (int i = 0; i <= steps; i++)

    {

        glVertex2d(round(X), round(Y));

        X += x\_increment;

        Y += y\_increment;

    }

    glEnd();

    glBegin(GL\_POINTS);

    glVertex2d(0, 0);

    X0 = -100, Y0 = 0, X1 = 0, Y1 = 200;

    dx = X1 - X0;

    dy = Y1 - Y0;

    steps = abs(dx) > abs(dy) ? abs(dx) : abs(dy);

    x\_increment = dx / (float)steps;

    y\_increment = dy / (float)steps;

    X = X0;

    Y = Y0;

    for (int i = 0; i <= steps; i++)

    {

        glVertex2d(round(X), round(Y));

        X += x\_increment;

        Y += y\_increment;

    }

    glEnd();

    glBegin(GL\_POINTS);

    glVertex2d(0, 0);

    X0 = 100, Y0 = 0, X1 = 0, Y1 = -200;

    dx = X1 - X0;

    dy = Y1 - Y0;

    steps = abs(dx) > abs(dy) ? abs(dx) : abs(dy);

    x\_increment = dx / (float)steps;

    y\_increment = dy / (float)steps;

    X = X0;

    Y = Y0;

    for (int i = 0; i <= steps; i++)

    {

        glVertex2d(round(X), round(Y));

        X += x\_increment;

        Y += y\_increment;

    }

    glEnd();

    glBegin(GL\_POINTS);

    glVertex2d(0, 0);

    X0 = -100, Y0 = 0, X1 = 0, Y1 = -200;

    dx = X1 - X0;

    dy = Y1 - Y0;

    steps = abs(dx) > abs(dy) ? abs(dx) : abs(dy);

    x\_increment = dx / (float)steps;

    y\_increment = dy / (float)steps;

    X = X0;

    Y = Y0;

    for (int i = 0; i <= steps; i++)

    {

        glVertex2d(round(X), round(Y));

        X += x\_increment;

        Y += y\_increment;

    }

    glEnd();

    glFlush();

}

int main(int argc, char\* argv[]) {

    glutInit(&argc, argv);

    glutInitDisplayMode(GLUT\_SINGLE | GLUT\_RGB);

    glutInitWindowSize(600, 600);

    glutCreateWindow("DDA");

    glutDisplayFunc(myDDA);

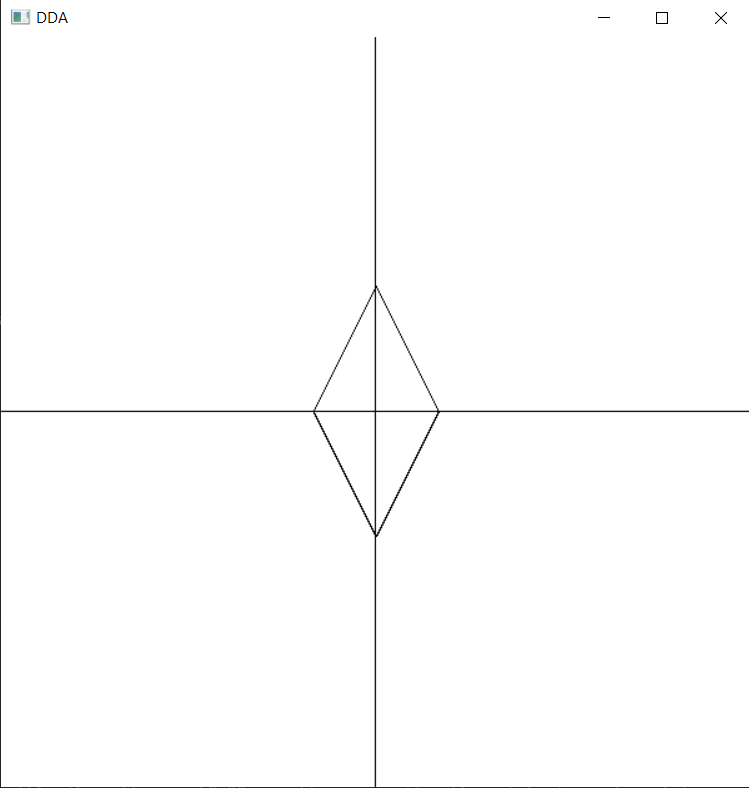
    myInit();

    glutMainLoop();

    return 1;

}

**OUTPUT:**

****