**UCS 1712 – GRAPHICS AND MULTIMEDIA LAB**

**ASSIGNMENT – 3**

**VISHAL N**

**185001198**

**04.08.2021 CSEC**

**1. BRESENHAM’S LINE DRAWING ALGORITHM:**

#include <GL/glut.h>

#include <stdio.h>

#include <iostream>

using namespace std;

int x\_1, y\_1, x\_2, y\_2;

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 draw\_pixel(int x, int y) {

    glBegin(GL\_POINTS);

    glVertex2i(x, y);

    glEnd();

}

void myBresenham(int x\_1, int x\_2, int y\_1, int y\_2) {

    int dx, dy, i, e;

    int incx, incy, inc1, inc2;

    int x,y;

    dx = x\_2 - x\_1;

    dy = y\_2 - y\_1;

    if (dx < 0) dx = -dx;

    if (dy < 0) dy = -dy;

    incx = 1;

    if (x\_2 < x\_1) incx = -1;

    incy = 1;

    if (y\_2 < y\_1) incy = -1;

    x = x\_1; y = y\_1;

    if (dx > dy) {

        draw\_pixel(x, y);

        e = 2 \* dy-dx;

        inc1 = 2\*(dy-dx);

        inc2 = 2\*dy;

        for (i=0; i<dx; i++) {

            if (e >= 0) {

                y += incy;

                e += inc1;

            }

            else

                e += inc2;

            x += incx;

            draw\_pixel(x, y);

        }

    } else {

        draw\_pixel(x, y);

        e = 2\*dx-dy;

        inc1 = 2\*(dx-dy);

        inc2 = 2\*dx;

        for (i=0; i<dy; i++) {

            if (e >= 0) {

                x += incx;

                e += inc1;

            }

            else

                e += inc2;

            y += incy;

            draw\_pixel(x, y);

        }

    }

}

void myDisplay() {

    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();

    myBresenham(x\_1, x\_2, y\_1, y\_2);

    glFlush();

}

int main(int argc, char \*\*argv) {

    cout << "Enter (x\_1, y\_1, x\_2, y\_2)" << endl;

    cin >> x\_1 >> y\_1 >> x\_2 >> y\_2;

    glutInit(&argc, argv);

    glutInitDisplayMode(GLUT\_SINGLE|GLUT\_RGB);

    glutInitWindowSize(600, 600);

    glutInitWindowPosition(0, 0);

    glutCreateWindow("Bresenham");

    myInit();

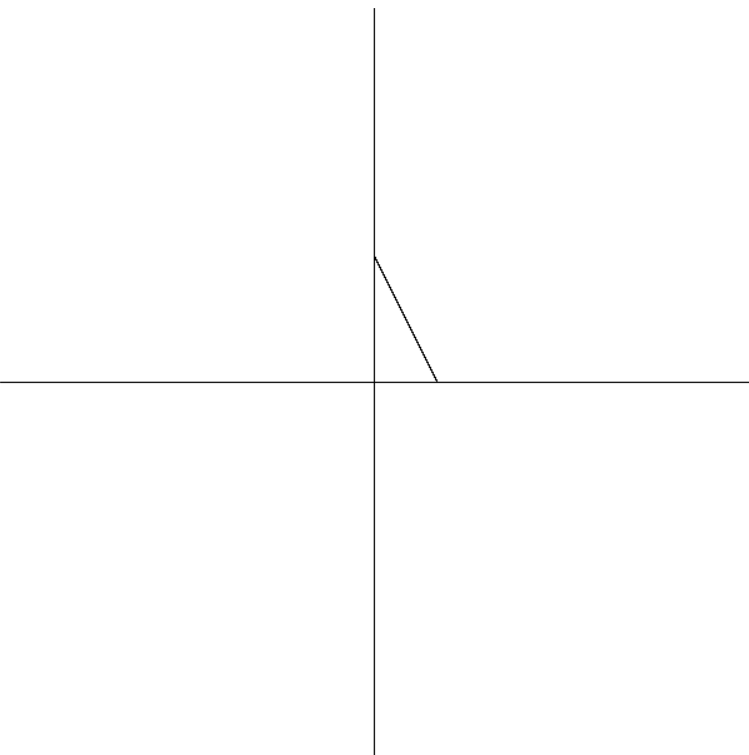
    glutDisplayFunc(myDisplay);

    glutMainLoop();

    return 0;

}

**OUTPUT:**

****