```c
/*
* Program : bresenham.cpp
*/
#include <GL/glut.h>

void init() {
    glClearColor(0.0, 0.0, 0.0, 0.0);
    glMatrixMode(GL_PROJECTION);
    glLoadIdentity();
    gluOrtho2D(0.0, 500.0, 0.0, 500.0);
    glColor3f(1.0, 1.0, 1.0);
}

void draw_pixel(int x, int y) {
    glBegin(GL_POINTS);
        glVertex2i(x, y);
    glEnd();
}

void generateByBresenham(int x1, int x2, int y1, int y2) {
    int incx = 1, incy = 1;

    int dx = x2 - x1;
    int dy = y2 - y1;

    if (dx < 0) dx = -dx;
    if (dy < 0) dy = -dy;

    if (x2 < x1) incx = -1;
    if (y2 < y1) incy = -1;

    int x = x1;
    int y = y1;

    // p = decision parameter

    if (dx > dy) {
        // Slope is less than 1
        draw_pixel(x, y);
        int p = 2 * dy - dx;
        for (int i = 0; i < dx; i++) {
            if (p >= 0) {
                y += incy;
                p += 2 * (dy - dx);
            } else {
                p += 2 * dy;
            }
            x += incx;
            draw_pixel(x, y);
        }

    } else {
        // Slope is greater than 1
        draw_pixel(x, y);
        int p = 2 * dx - dy;
        for (int i = 0; i < dy; i++) {
            if (p >= 0) {
                x += incx;
                p += 2 * (dx - dy);;
            }
            else{
                p += 2 * dx;
            }
            y += incy;
            draw_pixel(x, y);
        }
```

```
    }
}

void display() {
    generateByBresenham(100, 400, 400, 200);
    generateByBresenham(100, 400, 400, 400);
    glFlush();
}


int main(int argc, char **argv) {
    glutInit(&argc, argv);
    glutInitDisplayMode(GLUT_SINGLE | GLUT_RGB | GLUT_DEPTH);
    glutInitWindowSize(500, 500);
    glutInitWindowPosition(300, 200);
    glutCreateWindow("Line generation using Bresenham's LGA");
    init();
    glutDisplayFunc(display);
    glutMainLoop();
    return 0;
}
```