```
#include<OpenGL/gl.h>
#include <GLUT/glut.h>
#include <math.h>

#include <iostream>
using namespace std;

struct Point {
	GLint x;
	GLint y;
};

int Abs(int x) {

	if(x<0)
		return (x*(-1));
	else
		return x;
}

Point p1, p2;

void draw_dda(Point p1, Point p2) {
	GLfloat dx = p2.x - p1.x;
	GLfloat dy = p2.y - p1.y;

	GLfloat x1 = p1.x;
	GLfloat y1 = p1.y;

	GLfloat step = 0;

	if(Abs(dx) > Abs(dy)) {
		step = Abs(dx);
	} else {
		step = Abs(dy);
	}

	GLfloat xInc = dx/step;
	GLfloat yInc = dy/step;

	glClear(GL_COLOR_BUFFER_BIT);
	glBegin(GL_POINTS);
	for(float i = 1; i <= step; i++) {
		glVertex2i(x1, y1);
		x1 += xInc;
		y1 += yInc;
	}
	glEnd();
	glFlush();
}

void myMouseFunc(int button, int state, int x, int y)
{
	if(button == GLUT_LEFT_BUTTON && state == GLUT_DOWN) {
		p1.x = x;
		p1.y = 480 - y;
	}
	else if(button == GLUT_LEFT_BUTTON && state == GLUT_UP) {
		p2.x = x;
		p2.y = 480 - y;
```

```
                draw_dda(p1, p2);
        }
}

void init() {
        glClearColor(1.0, 1.0, 1.0, 0.0);
        glColor3f(0.0f, 0.0f, 0.0f);
        glPointSize(1.0f);
        gluOrtho2D(0.0f, 640.0f, 0.0f, 480.0f);
}

void display(void) {}

int main(int argc, char **argv) {
        glutInit(&argc, argv);
        glutInitDisplayMode(GLUT_SINGLE|GLUT_RGB);
        glutInitWindowPosition(200, 200);
        glutInitWindowSize(640, 480);
        glutCreateWindow("Mouse Func");
        glutDisplayFunc(display);
        glutMouseFunc(myMouseFunc);
        init();
        glutMainLoop();

        return 0;
}
```