```
// PRESENTADO POR JOAN SEBASTIAN TIBAQUIRA COD 1202060
// PRESENTADO POR JOAN SEBASTIAN TIBAQUIRA COD 1202060
#define GL_SILENCE_DEPRECATION
#include <cmath>
#include <GL/glut.h>
int W = 640, H = 640;
int cx, cy, cx1, cy1;
void Start() {
	glFlush();
}
void Pixel(int X, int Y) {
	glPointSize(10);//ADECUA EL TAMAÑO DEL PIXEL
	glColor3f(1, 0, 0);
	glBegin(GL_POINTS);
	glVertex2f(X, Y);
	glEnd();

}
float Minor(float a, float b) {
	float minor;
	if (a < b) {
		minor = a;
	}
	else {
		minor = b;
	}
	return minor;
}
void Line(int x1, int y1, int x2, int y2) {
	float m, b, dx, dy;
	if (x1 != x2) {
		dx = x2 - x1;
		dy = y2 - y1;
		m = dy / dx;
		b = y1 - (m * x1);
		if (m == 0) {
			for (int i = 0; i <= abs(x1 - x2); i++) {
				Pixel(i + Minor(x1, x2), y1);
			}
		}
		else {
			if (abs(m) == 1 || abs(y1 - y2) < abs(x1 - x2)) {
				for (int i = 0; abs(x1 - x2); i++) {
					Pixel(i + Minor(x1, x2), m * (i + Minor(x1, x2)) + b);
				}
			}
			else {
				for (int i = 0; i < abs(y1 - y2); i++) {
					Pixel((i + Minor(y1, y2) - b) / m, i + Minor(y1, y2));
				}
			}
		}
	}
	else {
		for (int i = Minor(y1, y2); i < abs(y1 - y2); i++) {
			Pixel(x1, i);
		}
	}
}
void Mouse(int B, int S, int X, int Y) {
       if ((S == GLUT_UP) && (B == GLUT_LEFT_BUTTON)) {
              cx = X;
              cy = Y;
              Pixel(cx, cy);
       }
       if ((S == GLUT_UP) && (B == GLUT_RIGHT_BUTTON)) {
              cx1 = X;
              cy1 = Y;
              Pixel(cx1, cy1);
              Line(cx, cy, cx1, cy1);
       }
}

int main(int argc, char* argv[]) {
       glutInit(&argc, argv);
       glutInitDisplayMode(GLUT_RGBA | GLUT_DOUBLE);
       glutInitWindowPosition(650, 50);
       glutInitWindowSize(W, H);
       glutCreateWindow("LINE");
       gluOrtho2D(0, W, H, 0);
       glClearColor(255, 255, 255, 0);
       glutDisplayFunc(Start);
       glutMouseFunc(Mouse);
       glutMainLoop();
}
/*ALGORITMO DDA
void DDA(float x1,float x2,float y1,float y2){
float L;
float dx,dy,cx,cy;
dx=(x2-x1);
dy=(y2-y1);
if(abs(dx)>=abs(dy)){
L=abs(dx);
}
else{
L=abs(dy);
}
cx=dx/L;
cy=dy/L;

for(int k=0;k<L;k++){
x1=x1+cx;
y1=y1+cy;
Pixel(round(x1),round(y1));
}
}
*/
```

```
#include <cmath>
#include <GL/glut.h>

int W = 640, H = 640;
int cx, cy, cx1, cy1;
void Start() {
        glFlush();
}
void Pixel(int X, int Y) {
        glPointSize(10);//ADECUA EL TAMAÑO DEL PIXEL
        glColor3f(1, 0, 0);
        glBegin(GL_POINTS);
        glVertex2f(X, Y);
        glEnd();

}
void DDA(float x1, float x2, float y1, float y2) {
        float L;
        float dx, dy, cx, cy;
        dx = (x2 - x1);
        dy = (y2 - y1);
        if (abs(dx) >= abs(dy)) {
                L = abs(dx);
        }
        else {
                L = abs(dy);
        }
        cx = dx / L;
        cy = dy / L;
        for (int k = 0; k < L; k++) {
                x1 = x1 + cx;
                y1 = y1 + cy;
                Pixel(round(x1), round(y1));
        }
}
void Mouse(int B, int S, int X, int Y) {
        if ((S == GLUT_UP) && (B == GLUT_LEFT_BUTTON)) {
                cx = X;
                cy = Y;
                Pixel(cx, cy);
        }
        if ((S == GLUT_UP) && (B == GLUT_RIGHT_BUTTON)) {
                cx1 = X;
                cy1 = Y;
                Pixel(cx1, cy1);
                DDA(cx, cy, cx1, cy1);
        }
}

int main(int argc, char* argv[]) {
        glutInit(&argc, argv);
        glutInitDisplayMode(GLUT_RGBA | GLUT_DOUBLE);
        glutInitWindowPosition(650, 50);
        glutInitWindowSize(W, H);
        glutCreateWindow("LINE");
        gluOrtho2D(0, W, H, 0);
        glClearColor(255, 255, 255, 0);
        glutDisplayFunc(Start);
```

```
       glutMouseFunc(Mouse);
       glutMainLoop();
}
```