

```

#include<iostream>
#include<GL/glut.h>
int contador = 0, detector = 0, final = 0;
int W = 500, H = 500;
int Matriz[500][500];
int Auxcolumna = 0;
int Vectorx[10], Vectory[10];
int figure = 0;
int i = 0, j = 0;
int Xo = 0, Yo = 0, Xf = 0, Yf = 0;
int num_Lineas = 0;
int xmenor, Aux_xmenor, ymenor, xmayor, Aux_xmayor, ymayor;
void Pintar_Pixel(float x, float y, int size, float r, float g, float b) {
    glBegin(GL_POINTS);
    glPointSize(size);
    glColor3f(r, g, b);
    glVertex2f(x, y);
    glEnd();
}
void PintaEjes() {
    int x, y;
    x = W / 2;
    y = 0;
    for (int i = 1; i < 1000; i++)//Eje Y
    {
        Pintar_Pixel(x, y, 1, 0, 0, 0);
        y++;
    }
    x = 0;
    y = H / 2;
    for (int j = 1; j < 1000; j++)// Eje X
    {
        Pintar_Pixel(x, y, 1, 0, 0, 0);
        x++;
    }
}
void PintarLinea(int x1, int y1, int x2, int y2) {
    num_Lineas++;
    int dx, dy, x, y;
    int Auxx, Auxy;
    double b, m;
    bool incremento, incremento2;
    dx = x2 - x1;
    dy = y2 - y1;
    x = x1;
    y = y1;
    if (x1 < x2) {
        incremento = true;
        Auxx = x2 - x1;
    }
    else {
        incremento = false;
        Auxx = x1 - x2;
    }
    if (y1 < y2) {
        incremento2 = true;
        Auxy = y2 - y1;
    }
}

```

```

else {
    incremento2 = false;
    Auxy = y1 - y2;
}

if (dx == 0) {
    for (int i = 0; i <= Auxy; i++) {
        Pintar_Pixel(x, y, 1, 0, 0, 0);
        Matriz[x][y] = 1;
        if (incremento2) {
            y++;
        }
        else
            y--;
    }
}
else
{
    if (dy == 0) {
        for (int i = 0; i <= Auxx; i++) {
            Pintar_Pixel(x, y, 1, 0, 0, 0);
            Matriz[x][y] = 1;
            if (incremento) {
                x++;
            }
            else
                x--;
        }
    }
    else {
        m = float(dy) / float(dx);
        b = y - (m * x);
        if (abs(dx) >= abs(dy)) {
            for (int i = 0; i <= Auxx; i++) {
                y = (m * x) + b;
                Pintar_Pixel(x, y, 1, 0, 0, 0);
                Matriz[x][y] = 1;
                if (incremento) {
                    x++;
                }
                else
                    x--;
            }
        }
        else {
            for (int i = 0; i <= Auxy; i++) {
                x = (y - b) / m;
                Pintar_Pixel(x, y, 1, 0, 0, 0);
                Matriz[x][y] = 1;
                if (incremento2) {
                    y++;
                }
                else
                    y--;
            }
        }
    }
}
}

```

```

}

void Rellenar() {
    for (int k = 0; k <= i; k++) {
        if (Vectorx[k] > xmayor)
            xmayor = Vectorx[k];
        Aux_xmayor = xmayor;
    }

    for (int m = 0; m <= j; m++) {
        if (Vectory[m] > ymayor)
            ymayor = Vectory[m];
    }
    xmenor = xmayor;
    for (int l = 0; l <= i; l++) {

        if (xmenor > Vectorx[l])
            xmenor = Vectorx[l];
        Aux_xmenor = xmenor;
    }
    ymenor = ymayor;
    for (int n = 0; n <= j; n++) {
        if (ymenor > Vectory[n])
            ymenor = Vectory[n];
        int xinicial_relleno, xfinal_relleno;
        int Puntos1 = 0, Puntos2 = 0;
        double Aux_R = 1, Aux_G = 1, Aux_B = 1;
        for (; ymenor <= ymayor + 1; ymenor++) {
            for (; xmenor <= xmayor; xmenor++) {
                if (Matriz[xmenor][ymenor] == 1 && Matriz[xmenor + 1][ymenor]
== 0) {

                    xinicial_relleno = xmenor + 1;
                    Puntos1 = 1;
                }
                else {
                    if (Matriz[xmenor][ymenor] == 0 && Matriz[xmenor +
1][ymenor] == 1) {

                        xfinal_relleno = xmenor - 1;
                        Puntos2 = 1;
                    }
                }
                if (Puntos1 == 1 && Puntos2 == 1) {
                    for (int Aux = xinicial_relleno; Aux <= xfinal_relleno;
Aux++) {

                        Pintar_Pixel(Aux, ymenor, 1, 1, 0, 1);
                    }
                }
            }
            xmenor = Aux_xmenor;
        }
    }
}

void OnMouse(int button, int state, int x, int y)
{
    if (button == GLUT_LEFT_BUTTON && state == GLUT_DOWN) {
        if (detector == 0) {

```

```

        Xo = x; Yo = y;
        Vectorx[0] = Xo;
        Vectory[0] = Yo;
    }
    else {
        if (detector >= 1 && detector < 10) {
            Xf = x; Yf = y;
            Xo = Vectorx[i];
            Yo = Vectory[j];
            Vectorx[i + 1] = Xf;
            Vectory[j + 1] = Yf;
            PintarLinea(Xo, Yo, Xf, Yf);
        }
        i++;
        j++;
    }
    detector++;
}
glutPostRedisplay();
}
void Display() {
    PintaEjes();
    glFlush();
}
void Teclado(unsigned char letra, int x, int y)
{
    switch (letra)
    {
        case 'r':
        case 'R':
            Rellenar();
            break;
    }
    glutPostRedisplay();
}
int main(int argc, char** argv) {
    glutInit(&argc, argv);
    glutInitDisplayMode(GLUT_RGBA);
    glutInitWindowPosition(450, 100);
    glutInitWindowSize(W, H);
    glutCreateWindow("Pintar Línea");
    glClearColor(0.7, 1, 0.7, 0);
    glutMouseFunc(OnMouse);
    glutKeyboardFunc(Teclado);
    glClear(GL_COLOR_BUFFER_BIT);
    glutDisplayFunc(Display);
    gluOrtho2D(0, W, H, 0);
    glutMainLoop();
    return 0;
}

```