```cpp
#include <iostream>
#include <GL/glut.h>

using namespace std;
int A=0;
int cx1 = 0, cx2 = 0, cy1 = 0, cy2 = 0;
struct GLColor {
       GLfloat red;
       GLfloat green;
       GLfloat blue;
};

GLColor colors[7] = {
{ 0.0f, 0.0f, 0.0f }, // Negro
{ 1.0f, 0.0f, 0.0f }, // Rojo
{ 0.0f, 1.0f, 0.0f }, // Verde
{ 0.0f, 0.0f, 1.0f }, // Azul
{ 1.0f, 1.0f, 0.0f }, // Amarillo
{ 1.0f, 0.0f, 1.0f }, // Morado
{ 0.0f, 1.0f, 1.0f } // Turquesa
};

GLColor color = colors[1];
GLsizei ancho = 800, alto = 600;

void Ventana(GLsizei a, GLsizei b) {
       glClearColor(1.0, 1.0, 1.0, 1.0);
       glMatrixMode(GL_PROJECTION);
       glLoadIdentity();
       glOrtho(0, (GLdouble)a, 0, (GLdouble)b, -1, 1);

       glMatrixMode(GL_MODELVIEW);
       glLoadIdentity();
       glViewport(0, 0, a, b);
       ancho = a;
       alto = b;
}

void PintaPixel(int x, int y) {
       glPointSize(10);
       glBegin(GL_POINTS);
       glColor3f(color.red, color.green, color.blue);
       glVertex2d(x, y);
       glEnd();
       glFlush();
}

void bres(int x1, int y1, int x2, int y2)
{
       int dx, dy, i, e;
       int incx, incy, inc1, inc2;
       int x, y;
       dx = cx2 - cx1;
       dy = cy2 - cy1;
       if (dx < 0) dx = -dx;
       if (dy < 0) dy = -dy;
       incx = 1;
       if (cx2 < cx1) incx = -1;
```

```
    incy = 1;
    if (cy2 < cy1) incy = -1;
    x = cx1;
    y = cy1;
    if (dx > dy)
    {
        PintaPixel(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;
            PintaPixel(x, y);
        }
    }
    else
    {
        PintaPixel(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;
            PintaPixel(x, y);
        }
    }
}

float round_value(float v)
{
    return floor(v + 0.5);
}

void LineaDDA(int x1, int y1, int x2, int y2)
{
    double dx = (cx2 - cx1);
    double dy = (cy2 - cy1);
    double steps;
    float xInc, yInc, x = cx1, y = cy1;
    steps = (abs(dx) > abs(dy)) ? (abs(dx)) : (abs(dy)); //Averiguar si incrementa en x o y
    xInc = dx / (float)steps;
    yInc = dy / (float)steps;
    glClear(GL_COLOR_BUFFER_BIT);  //Borrar pantalla
    glBegin(GL_POINTS); // Traza primer punto
```

```
        glVertex2d(x, y);
        int k;
        for (k = 0; k < steps; k++)
        {
                x += xInc;
                y += yInc;
                glVertex2d(round_value(x), round_value(y));
        }
        glEnd();
        glFlush();
}

void Display(void) {
        glFlush();
}

void raton(int btn, int state, int x, int y) {
        y = alto - y;
        if (btn == GLUT_LEFT_BUTTON && state == GLUT_DOWN) {
                cx1 = x;
                cy1 = y;
                PintaPixel(cx1, cy1);
        }
        if (btn == GLUT_RIGHT_BUTTON && state == GLUT_DOWN) {
                cx2 = x;
                cy2 = y;
                PintaPixel(cx2, cy2);
        }
}

void mainMenuHandler(int choice) {
        switch (choice) {
        case 1:// Bresenham
                bres(cx1, cy1, cx2, cy2);
                break;
        case 2:// DDA
                LineaDDA(cx1, cy1, cx2, cy2);
                break;
        case 3:
                glClear(GL_COLOR_BUFFER_BIT);
                glFlush();
                break;
        case 4:// Exit
                exit(0);
                break;
        }
}

void subMenuHandler(int choice) {
        color = colors[choice];
}

int main(int argc, char** argv) {

        glutInit(&argc, argv);
        glutInitDisplayMode(GLUT_SINGLE | GLUT_RGB);
        glutInitWindowSize(800, 600);
        glutInitWindowPosition(300, 150);
```

```
glutCreateWindow("LINEA (BRESENHAM - DDA)");
glutDisplayFunc(Display);
glutMouseFunc(raton);
glutReshapeFunc(Ventana);

int subMenu = glutCreateMenu(subMenuHandler);
glutAddMenuEntry("Negro", 0);
glutAddMenuEntry("Rojo", 1);
glutAddMenuEntry("Verde", 2);
glutAddMenuEntry("Azul", 3);
glutAddMenuEntry("Amarillo", 4);
glutAddMenuEntry("Morado", 5);
glutAddMenuEntry("Turquesa", 6);

glutCreateMenu(mainMenuHandler);
glutAddSubMenu("Cambiar Color", subMenu);
glutAddMenuEntry("Linea Bresenham", 1);
glutAddMenuEntry("Linea DDA", 2);
glutAddMenuEntry("Borrar Pantalla", 3);
glutAddMenuEntry("Exit", 4);

glutAttachMenu(GLUT_MIDDLE_BUTTON);
glutMainLoop();
return 0
```