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

struct Point {
	GLint x;
	GLint y;
};

struct Color {
	GLfloat r;
	GLfloat g;
	GLfloat b;
};

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;

	for(float i = 1; i <= step; i++) {
		glVertex2i(x1, y1);
		x1 += xInc;
		y1 += yInc;
	}
}

void init() {
	glClearColor(1.0, 1.0, 1.0, 0.0);
	glColor3f(0.0, 0.0, 0.0);
	glPointSize(1.0);
	glMatrixMode(GL_PROJECTION);
	glLoadIdentity();
	gluOrtho2D(0, 640, 0, 480);
}
```

```
Color getPixelColor(GLint x, GLint y) {
	Color color;
	glReadPixels(x, y, 1, 1, GL_RGB, GL_FLOAT, &color);
	return color;
}

void setPixelColor(GLint x, GLint y, Color color) {
	glColor3f(color.r, color.g, color.b);
	glBegin(GL_POINTS);
		glVertex2i(x, y);
	glEnd();
	glFlush();
}

void floodFill(GLint x, GLint y, Color oldColor, Color newColor) {
	Color color;
	color = getPixelColor(x, y);

	if(color.r == oldColor.r && color.g == oldColor.g && color.b == oldColor.b)
	{
		setPixelColor(x, y, newColor);
		floodFill(x+1, y, oldColor, newColor);
	   floodFill(x, y-1, oldColor, newColor);
		floodFill(x, y+1, oldColor, newColor);
		floodFill(x-1, y, oldColor, newColor);

	}
	return;
}

void onMouseClick(int button, int state, int x, int y)
{
	Color newColor = {1.0f, 0.0f, 0.0f};
	Color oldColor = {1.0f, 1.0f, 1.0f};

	floodFill(101, 299, oldColor, newColor);
}

void display(void) {
	Point p1 = {50, 50}, // bottom-left
      p2 = {300, 50}, // bottom-right (increased width)
      p3 = {300, 300}, // top-right (increased height)
      p4 = {50, 300}; // top-left (increased width and height)
	glClear(GL_COLOR_BUFFER_BIT);
	glBegin(GL_POINTS);
		draw_dda(p1, p2);
		draw_dda(p2, p3);
		draw_dda(p3, p4);
		draw_dda(p4, p1);
```

```
	glEnd();
	glFlush();
}

int main(int argc, char** argv)
{
	glutInit(&argc, argv);
	glutInitDisplayMode(GLUT_SINGLE|GLUT_RGB);
	glutInitWindowSize(640, 480);
	glutInitWindowPosition(400, 400);
	glutCreateWindow("Open GL");
	init();
	glutDisplayFunc(display);
	glutMouseFunc(onMouseClick);
	glutMainLoop();
	return 0;
}
```

OUTPUT: