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

int W = 800,H = 600;
int P[2][3];
int Hi = 0;

void Pixel(float x, float y) {
	glPointSize(2);
	glColor3f(0, 0, 1);
	glBegin(GL_POINTS);
	glVertex2f(x, y);
	glEnd();
	glFlush();

}
void Spline(int P[][3]) {
	for (float t = 0; t <= 1; t += 0.001)
	{
		Pixel(P[0][0] * pow(1 - t, 2) + P[0][1] * 2 * t * (1 - t) + P[0][2] *
pow(t, 2),
			P[1][0] * pow(1 - t, 2) + P[1][1] * 2 * t * (1 - t) + P[1][2] *
pow(t, 2));
	}

}
void Mouse(int btn, int state, int x, int y) {
	if (btn == GLUT_LEFT_BUTTON && state == GLUT_DOWN && Hi == 0)
	{
		P[0][0] = { x };
		P[1][0] = { y };
		Hi++;
		return;
	}
	if (btn == GLUT_LEFT_BUTTON && state == GLUT_DOWN && Hi == 1)
	{
		P[0][1] = { x };
		P[1][1] = { y };
		Hi++;
		return;
	}
	if (btn == GLUT_LEFT_BUTTON && state == GLUT_DOWN && Hi == 2)
	{
		P[0][2] = { x };
		P[1][2] = { y };
		Spline(P);
		Hi = 0;
		return;
	}

}
void vis() {
	glFlush();
}

int main(int argc, char** argv) {
	glutInit(&argc, argv);
```

```
    glutInitDisplayMode(GLUT_SINGLE | GLUT_RGBA);
    glutInitWindowSize(W, H);
    glutInitWindowPosition(0, 0);
    glutCreateWindow("Curvas Spline");
    glutDisplayFunc(vis);
    glutMouseFunc(Mouse);
    gluOrtho2D(0, W, H, 0);
    glClearColor(1, 1, 1, 1);
    glutMainLoop();
    return 0;
}
```