```
#include <GL/glut.h>
#include<iostream>
#include <GLFW/glfw3.h>

GLfloat m = 0.0;              // slope
GLfloat c = (-0 - m * -1);      // constant
GLfloat j = -0;
GLfloat i = 0;

static float rotAngle = 0.;

void init() {

}

void desenha() {
    glEnable(GL_POINT_SMOOTH);
    glEnable(GL_BLEND);
    glBlendFunc(GL_SRC_ALPHA, GL_ONE_MINUS_SRC_ALPHA);
    glHint(GL_POINT_SMOOTH_HINT, GL_FASTEST);
    glLineWidth(1.5);

    glClearColor(0.0, 0.0, 0.0, 0.0);

    glClear(GL_COLOR_BUFFER_BIT);

    glEnable(GLUT_MULTISAMPLE);

    m = (-0.5) / 2;

    glColor3f(1, 0, 0);
    glPointSize(5);
    GLfloat x = 0.0;
    glBegin(GL_POINTS);
    c += -0.5;
    for (double i = -1; i < 1; i += 0.001) {
        j = m * i * c;         // result of y from line function
        glVertex2f(i, j);
    }
    glEnd();
    glFlush();

}
```

```
void reshape(int w, int h)
{
    glViewport(0, 0, w, h);
    glMatrixMode(GL_PROJECTION);
    glLoadIdentity();
    if (w <= h)
        gluOrtho2D(-1.0, 1.0,
            -1.0 * (GLfloat)h / (GLfloat)w, 1.0 * (GLfloat)h /
(GLfloat)w);
    else
        gluOrtho2D(-1.0 * (GLfloat)w / (GLfloat)h,
            1.0 * (GLfloat)w / (GLfloat)h, -1.0, 1.0);
    glMatrixMode(GL_MODELVIEW);
    glLoadIdentity();
}

void keyboard(unsigned char key, int x, int y)
{
    switch (key) {
    case 'r':
    case 'R':
        rotAngle += 20.;
        if (rotAngle >= 360.) rotAngle = 0.;
        glutPostRedisplay();
        break;
    case 'a':
        glDisable(GL_POINT_SMOOTH);
        glDisable(GL_BLEND);
        break;
    case 27:  /*  Escape Key  */
        exit(0);
        break;
    default:
        break;
    }
}

int main(int argc, char** argv) {
    glutInit(&argc, argv);
    glutInitDisplayMode(GLUT_SINGLE | GLUT_RGB);
    glutInitWindowSize(800, 500);
    glutInitWindowPosition(50, 50);
    init();
    glutReshapeFunc(reshape);
    glutCreateWindow("Pontos");
    glutDisplayFunc(desenha);
    glutKeyboardFunc(keyboard);
    glutMainLoop();
    return 0;
}
```