Name: Bresenham's Line drawing algorithm derivation (GL_POINTS).

Code:

```
#include <windows.h>
#include <GL/glut.h>
#include <stdio.h>
#include <GL/gl.h>

int x1, y1, x2, y2, dx, dy, pk;

void init(void)
{
   glClearColor(0.0, 0.0, 0.0, 0.0);
   glMatrixMode(GL_PROJECTION);
   glLoadIdentity();
   gluOrtho2D(0.0, 1000.0, 0.0, 1000.0);
}

void Draw()
{
   glClear(GL_COLOR_BUFFER_BIT);
   glColor3f(1, 1, 1);
   //glPointSize(3.0);
   dy = y2 - y1;
   dx = x2 - x1;
   int x = x1;
   int y = y1;
   pk = 2 * dy - dx;

   glBegin(GL_POINTS);
   glVertex2i(x, y);

   for (int i=0; i<dx; i++)
   {
      if (pk < 0)
      {
```

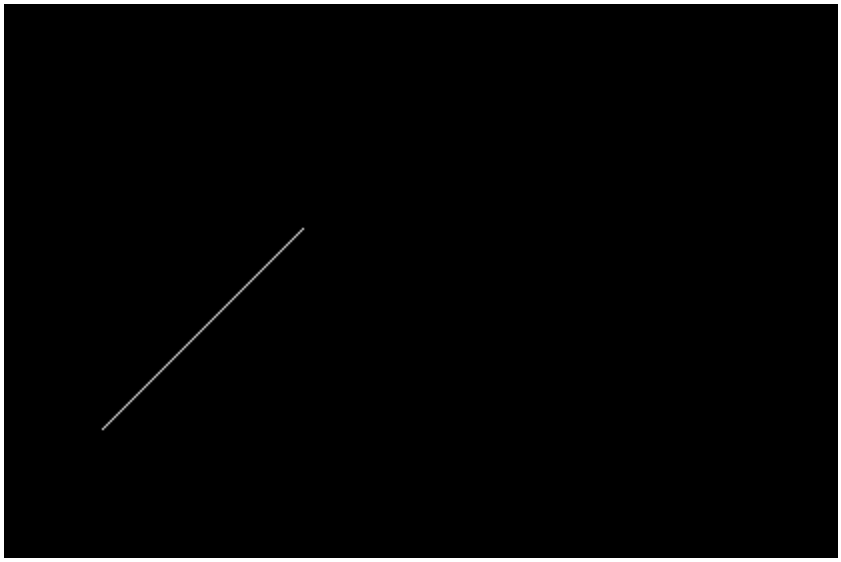
```
            x = x + 1;
            y = y;
            pk = pk + 2 * dy;
        }
        else
        {
            x = x + 1;
            y = y + 1;
            pk = pk + 2*dy - 2*dx;
        }
        glVertex2i(x, y);
    }
    glEnd();

    glutSwapBuffers();
}

int main(int argc, char **argv)
{
    printf("Enter x1 and y1: ");
    scanf("%d %d", &x1, &y1);
    printf("Enter x2 and y2: ");
    scanf("%d %d", &x2, &y2);

    glutInit(&argc, argv);
    glutInitDisplayMode(GLUT_RGB | GLUT_DOUBLE);
    glutInitWindowPosition(100, 100);
    glutInitWindowSize(500, 500);
    glutCreateWindow("Bresenham's Line Algorithm");
    init();
    glutDisplayFunc(Draw);
    glutMainLoop();
    return 0;
}
```

Output:



Name: Kite drawing using GL_LINES.

Code:

```
#include <windows.h>
#include <GL/glut.h>
#include <stdio.h>
#include <GL/gl.h>

void init(void)
{
   glClearColor(1.0, 1.0, 1.0, 0.0);
   glMatrixMode(GL_PROJECTION);
   glLoadIdentity();
   gluOrtho2D(0.0, 500.0, 0.0, 500.0);
}

void Draw()
{
   glClear(GL_COLOR_BUFFER_BIT);
   glColor3f(0.0, 0.0, 0.0);

   glBegin(GL_LINES);


   glVertex2i(250, 400);
   glVertex2i(150, 250);
```

```
    glVertex2i(250, 400);
    glVertex2i(350, 250);

    glVertex2i(150, 250);
    glVertex2i(250, 100);
    glVertex2i(350, 250);
    glVertex2i(250, 100);


    glVertex2i(250, 400);
    glVertex2i(250, 100);

    glVertex2i(150, 250);
    glVertex2i(350, 250);

    glVertex2i(250, 100);
    glVertex2i(200, 50);

    glVertex2i(250, 100);
    glVertex2i(300, 50);


    glVertex2i(200, 50);
    glVertex2i(300, 50);

    glEnd();
    glutSwapBuffers();
}

int main(int argc, char **argv)
{
    glutInit(&argc, argv);
    glutInitDisplayMode(GLUT_RGB | GLUT_DOUBLE);
    glutInitWindowPosition(100, 100);
    glutInitWindowSize(500, 500);
    glutCreateWindow("Basic Drawing");
```

```
    init();
    glutDisplayFunc(Draw);
    glutMainLoop();
    return 0;
}
```

Output: