ASSIGNMENT-3

1)
```
#include <iostream>
#include <GL/glut.h>

using namespace std;

void display(void)
{
    glClear(GL_COLOR_BUFFER_BIT);
    glColor3f(1.0, 1.0, 1.0);
    glPointSize(2.0);

    int x1 = 100, y1 = 100, x2 = 500, y2 = 500;
   float dx = x2 - x1;
    float dy = y2 - y1;
    float steps;

    if(abs(dx) > abs(dy))
        steps = abs(dx);
    else
        steps = abs(dy);

    float xIncrement = dx / steps;
   float yIncrement = dy / steps;
    float x = x1, y = y1;

    glBegin(GL_POINTS);
    for(int i = 0; i < steps; i++)
    {
        glVertex2i(x, y);
        x += xIncrement;
        y += yIncrement;
    }
    glEnd();

    glFlush();
}

void init(void)
{
    glClearColor(0.0, 0.0, 0.0, 0.0);
    glMatrixMode(GL_PROJECTION);
    glLoadIdentity();
    gluOrtho2D(0.0, 640.0, 0.0, 480.0);
}
```

```
int main(int argc, char** argv)
{
    glutInit(&argc, argv);
    glutInitDisplayMode(GLUT_SINGLE | GLUT_RGB);
    glutInitWindowSize(640, 480);
    glutInitWindowPosition(100, 150);
    glutCreateWindow("DDA Line Drawing");
    init();
    glutDisplayFunc(display);
    glutMainLoop();
    return 0;
}
```

2)

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

using namespace std;

void display(void)
{
    glClear(GL_COLOR_BUFFER_BIT);
    glColor3f(1.0, 1.0, 1.0);
    glPointSize(2.0);

    int x1 = 100, y1 = 100, x2 = 500, y2 = 500;
    int dx = abs(x2 - x1);
    int dy = abs(y2 - y1);
    int sx = (x1 < x2) ? 1 : -1;
    int sy = (y1 < y2) ? 1 : -1;
    int err = dx - dy;
    int x = x1, y = y1;

    glBegin(GL_POINTS);
    while(true)    {
        glVertex2i(x, y);
        if (x == x2 && y == y2) break;
       int e2 = 2 * err;
        if (e2 > -dy)
        {
            err -= dy;
            x += sx;
        }
        if (e2 < dx)
        {
            err += dx;
            y += sy;
        }
    }
    glEnd();
```

```
    glFlush();
}

void init(void)
{
    glClearColor(0.0, 0.0, 0.0, 0.0);
    glMatrixMode(GL_PROJECTION);
    glLoadIdentity();
    gluOrtho2D(0.0, 640.0, 0.0, 480.0);
}

int main(int argc, char** argv)
{
    glutInit(&argc, argv);
    glutInitDisplayMode(GLUT_SINGLE | GLUT_RGB);
    glutInitWindowSize(640, 480);
    glutInitWindowPosition(100, 150);
    glutCreateWindow("Bresenham's Line Drawing");
    init();
    glutDisplayFunc(display);
    glutMainLoop();
    return 0;
}
```