Practical no:02
Title:DDA Line drawing algorithm

Name:-Ujawala Sinha
Roll no:- S554

Code:

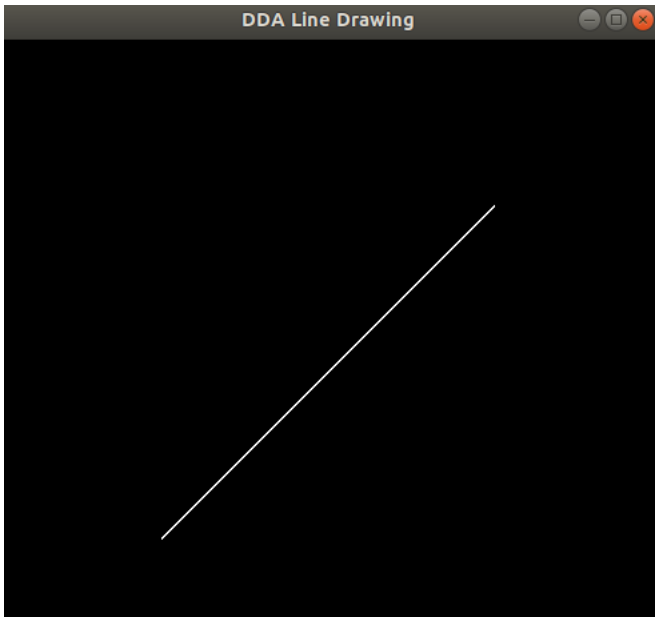
```
#include <GL/glut.h>
#include <iostream>

int lineType = 1;  // Default to Simple Line

// Function to draw a simple line using DDA algorithm
void drawSimpleLine() {
    glBegin(GL_LINES);
    glVertex2f(-0.5f, -0.5f);
    glVertex2f(0.5f, 0.5f);
    glEnd();
}

// Function to draw a dotted line
void drawDottedLine() {
    glEnable(GL_LINE_STIPPLE);
    glLineStipple(1, 0x0101);  // Dotted pattern (1 pixel on, 1 pixel off)
    glBegin(GL_LINES);
    glVertex2f(-0.5f, -0.5f);
    glVertex2f(0.5f, 0.5f);
    glEnd();
    glDisable(GL_LINE_STIPPLE);
}

// Function to draw a dashed line
void drawDashedLine() {
    glEnable(GL_LINE_STIPPLE);
    glLineStipple(1, 0x00FF);  // Dashed pattern (short dashes)
    glBegin(GL_LINES);
    glVertex2f(-0.5f, -0.5f);
    glVertex2f(0.5f, 0.5f);
    glEnd();
    glDisable(GL_LINE_STIPPLE);
}

// Function to initialize OpenGL settings
void initOpenGL() {
    glClearColor(0.0f, 0.0f, 0.0f, 1.0f);  // Set background color to black
    glColor3f(1.0f, 1.0f, 1.0f);           // Set line color to white
    glLineWidth(2.0f);                     // Set line width
}

// Display function called by GLUT to render the scene
void display() {
    glClear(GL_COLOR_BUFFER_BIT);  // Clear the screen
```

```
    // Draw the selected line type
    if (lineType == 1) {
        drawSimpleLine();
    } else if (lineType == 2) {
        drawDottedLine();
    } else if (lineType == 3) {
        drawDashedLine();
    }

    glFlush();  // Flush the OpenGL buffers
}

// Function to get user input before starting OpenGL
void getUserChoice() {
    std::cout << "OpenGL Line Drawing - Choose a line type:" << std::endl;
    std::cout << "Press 1: Simple Line" << std::endl;
    std::cout << "Press 2: Dotted Line" << std::endl;
    std::cout << "Press 3: Dashed Line" << std::endl;
    std::cout << "Enter your choice: ";

    int choice;
    std::cin >> choice;

    if (choice >= 1 && choice <= 3) {
        lineType = choice;
    } else {
        std::cout << "Invalid choice. Defaulting to Simple Line." << std::endl;
        lineType = 1;
    }
}

// Main function to set up the window and start the main loop
int main(int argc, char** argv) {
    getUserChoice();  // Prompt the user before opening the window

    glutInit(&argc, argv);
    glutInitDisplayMode(GLUT_SINGLE | GLUT_RGB);  // Single buffer and RGB color mode
    glutInitWindowSize(500, 500);                 // Set window size
    glutCreateWindow("DDA Line Drawing");         // Create the window
    initOpenGL();                                 // Initialize OpenGL settings
    glutDisplayFunc(display);                     // Register display function
    glutMainLoop();                               // Start the GLUT main loop
    return 0;
}
```

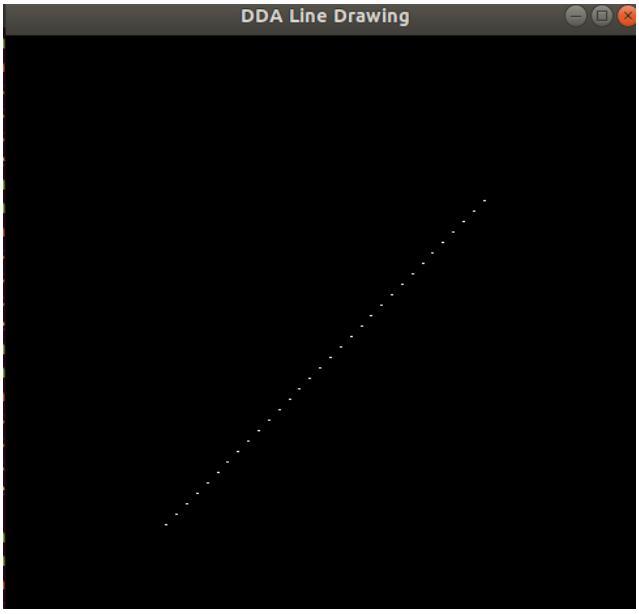
Output:

```
svpm@svpm-HP-EliteDesk-800-G2-SFF:~$ g++ ddau.cpp -lGL -lGLU -lglut
```

```
svpm@svpm-HP-EliteDesk-800-G2-SFF:~$ ./a.out
OpenGL Line Drawing - Choose a line type:
Press 1: Simple Line
Press 2: Dotted Line
Press 3: Dashed Line
Enter your choice: 1
```




```
svpm@svpm-HP-EliteDesk-800-G2-SFF:~$ g++ ddau.cpp -lGL -lGLU -lglut
svpm@svpm-HP-EliteDesk-800-G2-SFF:~$ ./a.out
OpenGL Line Drawing - Choose a line type:
Press 1: Simple Line
Press 2: Dotted Line
Press 3: Dashed Line
Enter your choice: 2
```



```
svpm@svpm-HP-EliteDesk-800-G2-SFF:~$ g++ ddau.cpp -lGL -lGLU -lglut
svpm@svpm-HP-EliteDesk-800-G2-SFF:~$ ./a.out
OpenGL Line Drawing - Choose a line type:
Press 1: Simple Line
Press 2: Dotted Line
Press 3: Dashed Line
Enter your choice: 3
```