Practical No.:2

Title: DDA and Bresenham line drawing algorithm .
Name: Samruddhi Sangram Yadav.
Roll No.:S564

Bresenham line drawing algorithm

```
#include <GL/glut.h>
#include <iostream>
using namespace std;
struct Point {
   GLint x, y;
};
int choice; // User choice for line style
void drawBresenham(Point p1, Point p2) {
   int dx = abs(p2.x - p1.x);
   int dy = abs(p2.y - p1.y);
   int sx = (p1.x < p2.x) ? 1 : -1;
   int sy = (p1.y < p2.y) ? 1 : -1;
   int err = dx - dy;
   int count = 0; // Counter to handle different patterns
   glBegin(GL_POINTS);
   while (true) {
      if (choice == 1) {
         // Simple Line
         glVertex2i(p1.x, p1.y);
      }
      else if (choice == 2 && count % 5 == 0) {
         // Dotted Line (draw every 5th point)
         glVertex2i(p1.x, p1.y);
      }
      else if (choice == 3 && (count / 5) % 2 == 0) {
         // Dashed Line (draw 5 pixels, skip 5)
         glVertex2i(p1.x, p1.y);
      }
      else if (choice == 4 && ((count / 5) % 3 == 0 || count % 5 == 0)) {
         // Dotted and Dashed Line (combination of both)
         glVertex2i(p1.x, p1.y);
      }
      if (p1.x == p2.x && p1.y == p2.y) break;
      int e2 = 2 * err;
      if (e2 > -dy) {
         err -= dy;
         p1.x += sx;
      }
      if (e2 < dx) {
         err += dx;
         p1.y += sy;
      }
```

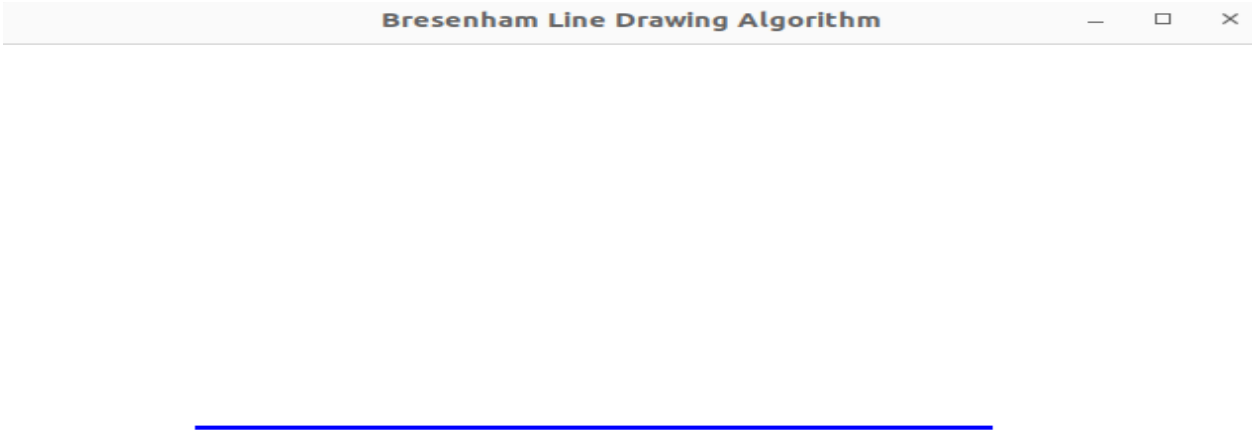
```
        count++; // Increment counter for line style patterns
    }
    glEnd();
    glFlush();
}
void display() {
    glClear(GL_COLOR_BUFFER_BIT);
    Point p1 = {100, 200}, p2 = {500, 200}; // Line coordinates
    glColor3f(0.0, 0.0, 1.0); // Blue color
    drawBresenham(p1, p2);
    glFlush();
}
void init() {
    glClearColor(1.0, 1.0, 1.0, 0.0);
    glPointSize(3.0);
    glMatrixMode(GL_PROJECTION);
    glLoadIdentity();
    gluOrtho2D(0, 640, 0, 480);
}
int main(int argc, char** argv) {
    // Get user choice in the terminal
    cout << "Select Line Type:\n";
    cout << "1: Simple Line\n";
    cout << "2: Dotted Line\n";
    cout << "3: Dashed Line\n";
    cout << "4: Dotted and Dashed Line\n";
    cout << "Enter your choice: ";
    cin >> choice;

    if (choice < 1 || choice > 4) {
        cout << "Invalid Choice! Exiting program." << endl;
        return 0; // Exit if choice is invalid
    }
    glutInit(&argc, argv);
    glutInitDisplayMode(GLUT_SINGLE | GLUT_RGB);
    glutInitWindowSize(640, 480);
    glutInitWindowPosition(100, 100);
    glutCreateWindow("Bresenham Line Drawing Algorithm");
    init();
    glutDisplayFunc(display);
    glutMainLoop();
    return 0;
}
```

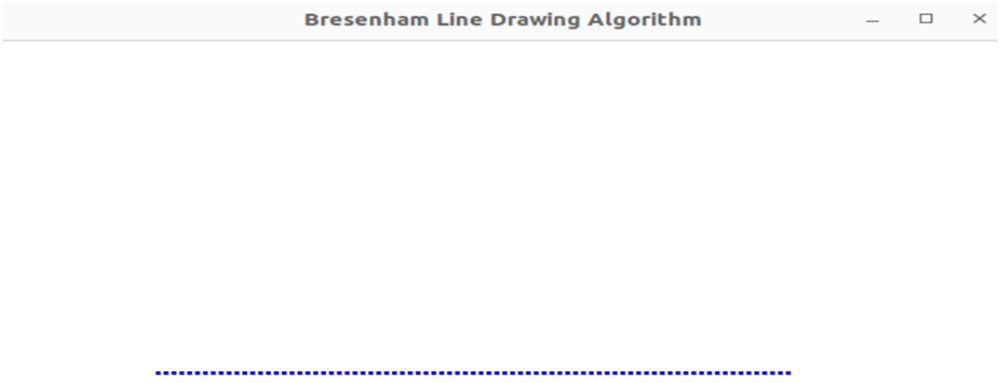
Output:

```
it@it-HP-EliteDesk-800-G2-SFF:~$ g++ bline1.cpp -lGL -lGLU -lglut
it@it-HP-EliteDesk-800-G2-SFF:~$ ./a.out
Select Line Type:
```

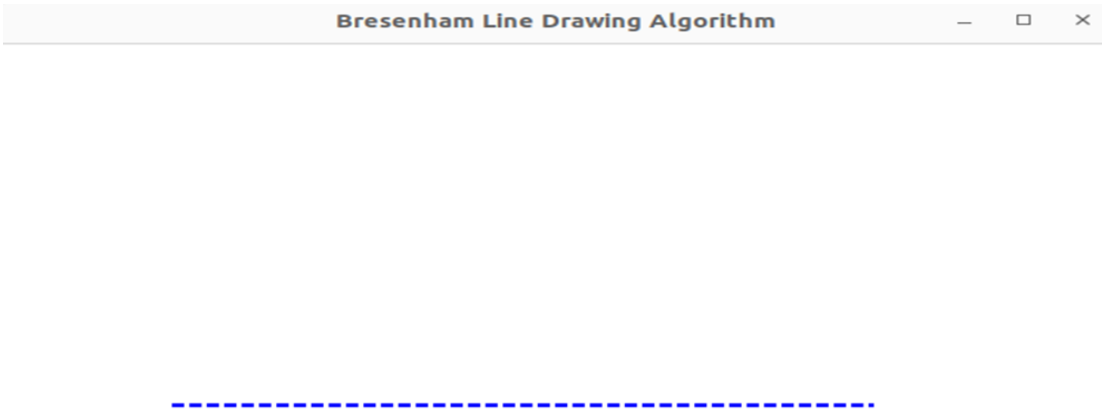
1: Simple Line
2: Dotted Line
3: Dashed Line
4: Dotted and Dashed Line
Enter your choice: 1



it@it-HP-EliteDesk-800-G2-SFF:~$ g++ bline1.cpp -lGL -lGLU -lglut
it@it-HP-EliteDesk-800-G2-SFF:~$ ./a.out
Select Line Type:
1: Simple Line
2: Dotted Line
3: Dashed Line
4: Dotted and Dashed Line
Enter your choice: 2



```
it@it-HP-EliteDesk-800-G2-SFF:~$ g++ bline1.cpp -lGL -lGLU -lglut
it@it-HP-EliteDesk-800-G2-SFF:~$ ./a.out
Select Line Type:
1: Simple Line
2: Dotted Line
3: Dashed Line
4: Dotted and Dashed Line
Enter your choice: 3
```




```
it@it-HP-EliteDesk-800-G2-SFF:~$ g++ bline1.cpp -lGL -lGLU -lglut
it@it-HP-EliteDesk-800-G2-SFF:~$ ./a.out
Select Line Type:
1: Simple Line
2: Dotted Line
3: Dashed Line
4: Dotted and Dashed Line
Enter your choice: 4
```