NATIONAL INSTITUTE OF TECHNOLOGY SILCHAR

Cachar, Assam

B.Tech. VIth Sem

Subject Code: CS-317

Subject Name: Graphics and Multimedia Lab

Submitted By:

Name : Subhojit Ghimire

Sch. Id. : 1912160

Branch : CSE – B

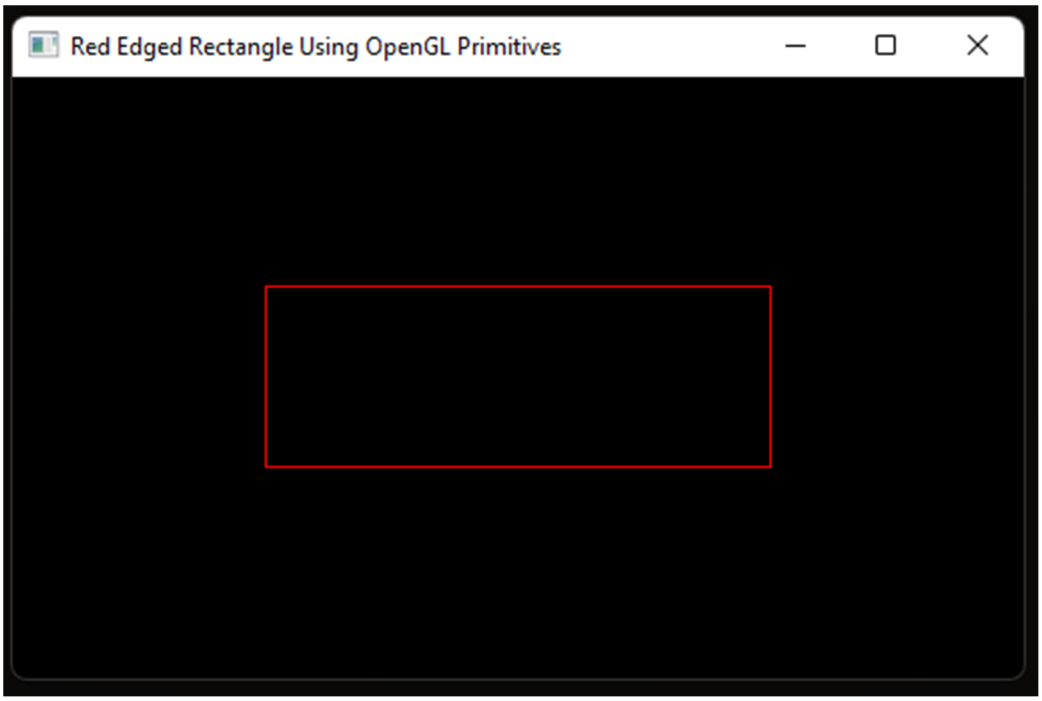
1. **Using OpenGL Primitives display a rectangular window placed it in suitable position and draw a rectangle inside the window having red edges.**

   ➔ **CODE:**

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

void initGL() {
}

void display() {
   glClear(GL_COLOR_BUFFER_BIT);

   glBegin(GL_LINE_LOOP);
      glColor3f(1.0f, 0.0f, 0.0f);
      glVertex2f(-0.5f, 0.3f);
      glVertex2f(-0.5f, -0.3f);
      glVertex2f(0.5f, -0.3f);
      glVertex2f(0.5f, 0.3f);
   glEnd();

   glFlush();
}

int main(int argc, char** argv) {
   glutInit(&argc, argv);
   glutCreateWindow("Red Edged Rectangle Using OpenGL Primitives");
   glutInitWindowSize(500, 500);
   glutInitWindowPosition(0, 0);
   glutDisplayFunc(display);
   initGL();
   glutMainLoop();
   return 0;
}
```

OUTPUT:

// Rectangle with Red Edge



2. **Implementation of DDA line drawing algorithm. "The program should take input of the initial end-point and the final end-point. Divide the coordinate axes into four quadrants and draw the line. Show that your program works for [m>0, m=0 and m<0]".**

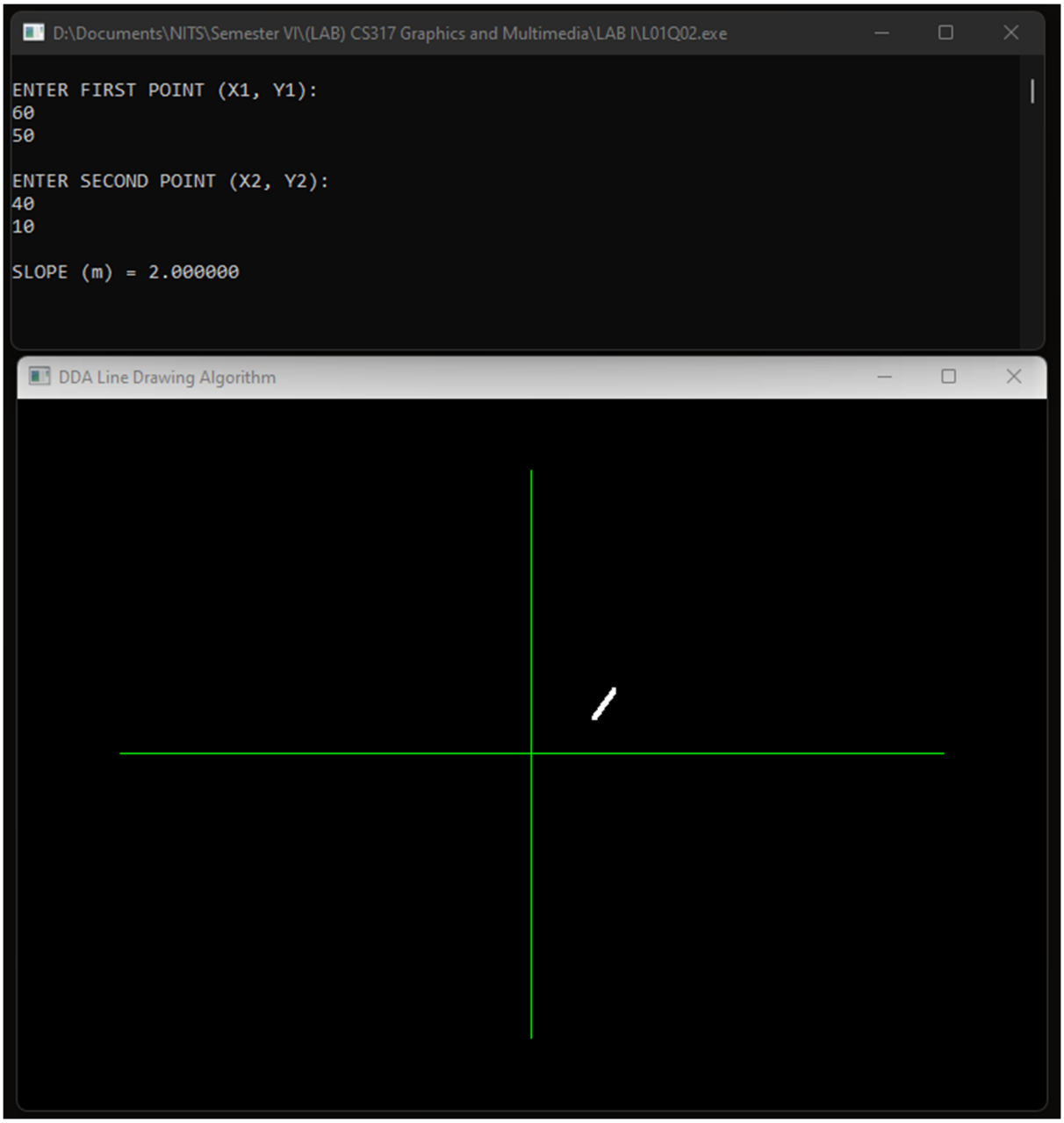
➔ **CODE:**

```
#include <stdio.h>
#include <math.h>
#include <GL/glut.h>

double X1, Y1, X2, Y2;

void LineDDA (void) {
    GLint dx = abs(X2 - X1);
    GLint dy = abs(Y2 - Y1);
    GLdouble steps;
    GLfloat xInc, yInc, xx = X1, yy = Y1;
    steps = (dx > dy) ? (dx) : (dy);
    xInc = dx/(GLfloat)steps;
    yInc = dy/(GLfloat)steps;

    glClear (GL_COLOR_BUFFER_BIT);

    glBegin (GL_POINTS);
        glColor3f (1.0, 1.0, 1.0);
        glVertex2d (xx, yy);
        int kk;
        for (kk = 0; kk < steps; ++kk) {
            xx += xInc;
            yy += yInc;
            glVertex2i (floor(xx + 0.5), floor(yy + 0.5));
        }
    glEnd();

    glBegin (GL_LINE_LOOP);
        glColor3f (0.0, 1.0, 0.0);
        glVertex2f (-400, 0);
        glVertex2f (400, 0);
    glEnd();

    glBegin (GL_LINE_LOOP);
        glColor3f (0.0, 1.0, 0.0);
        glVertex2f (0, -400);
        glVertex2f (0, 400);
    glEnd();

    glFlush();
}
```

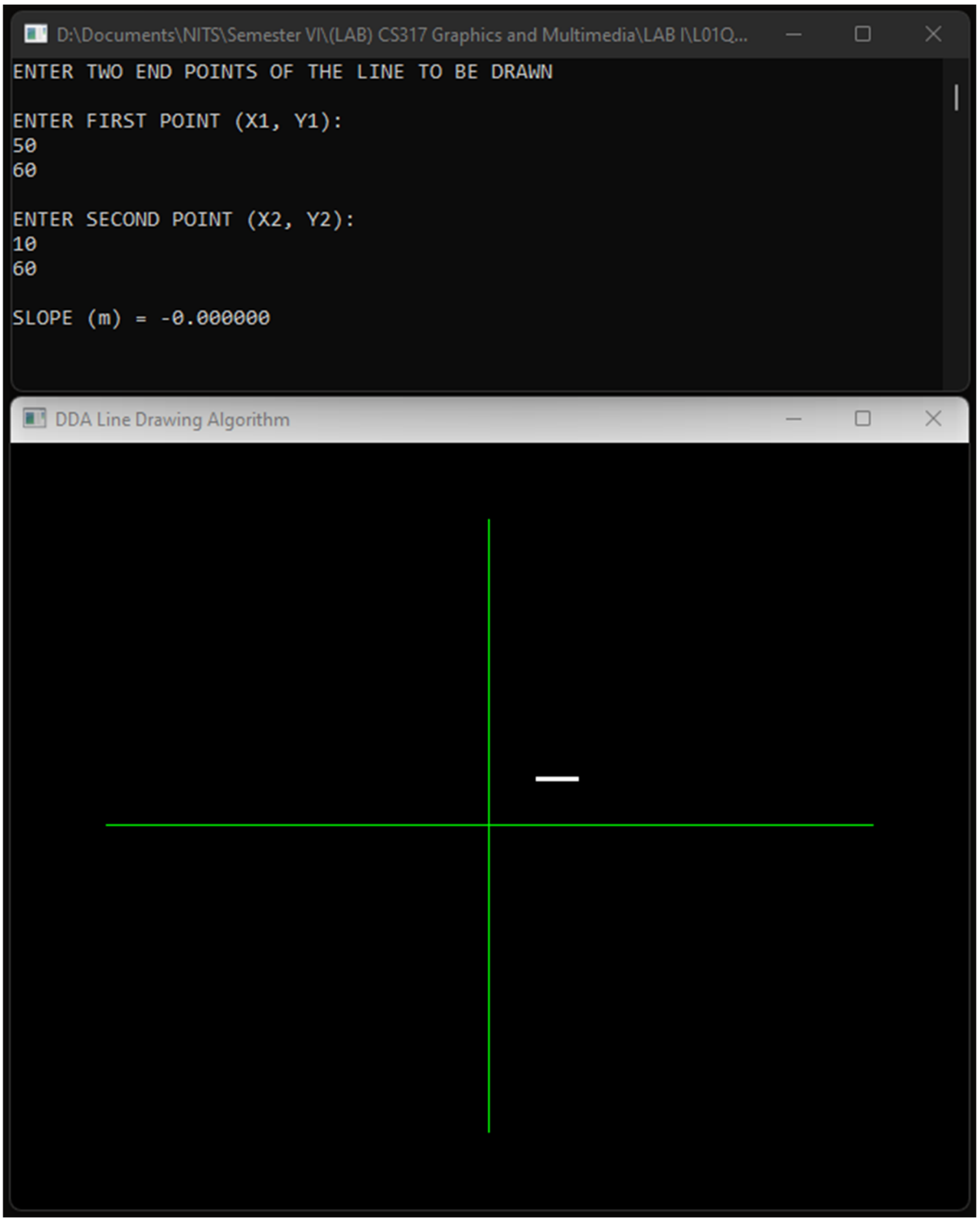
```
void Init () {
    glClearColor (0.0, 0.0, 0.0, 0.0);
    glColor3f (1.0, 1.0, 1.0);
    glPointSize (3.0);
    glMatrixMode (GL_PROJECTION);
    glLoadIdentity ();
    gluOrtho2D (-500, 500, -500, 500);
}

int main (int argc, char** argv) {
    printf ("ENTER TWO END POINTS OF THE LINE TO BE DRAWN\n");
    printf ("\nENTER FIRST POINT (X1, Y1):\n");
    scanf (" %lf %lf", &X1, &Y1);
    printf ("\nENTER SECOND POINT (X2, Y2):\n");
    scanf (" %lf %lf", &X2, &Y2);
    printf ("\nSLOPE (m) = %lf", (Y2-Y1)/(X2-X1));

    glutInit (&argc,argv);
    glutInitDisplayMode (GLUT_SINGLE | GLUT_RGB);
    glutInitWindowPosition (0, 0);
    glutInitWindowSize (500, 500);
    glutCreateWindow ("DDA Line Drawing Algorithm");
    glutDisplayFunc (LineDDA);
    Init ();
    glutMainLoop ();
    return 0;
}
```

OUTPUT:

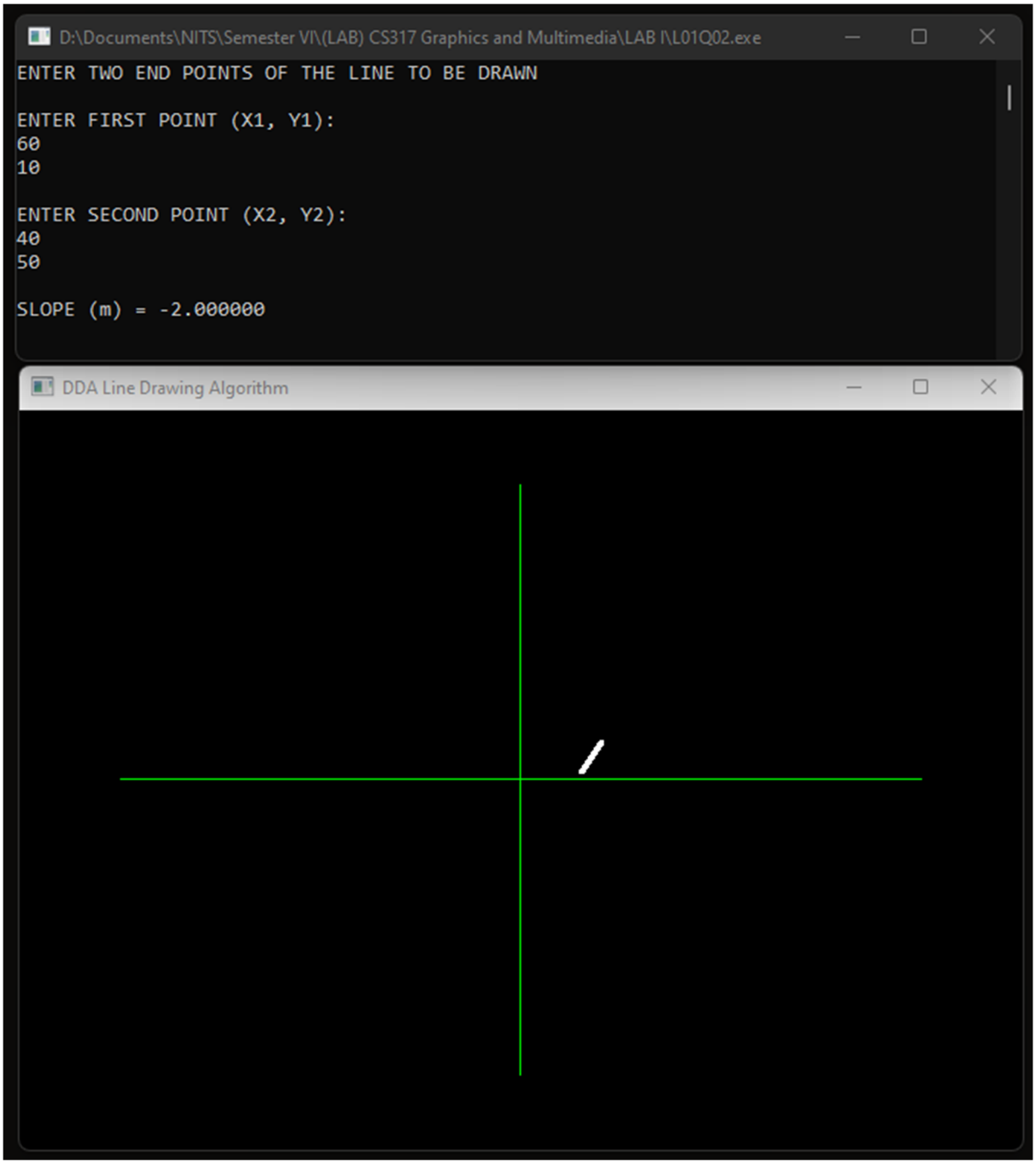
// m > 0


D:\Documents\NITS\Semester VI\(LAB) CS317 Graphics and Multimedia\LAB I\L01Q02.exe
ENTER FIRST POINT (X1, Y1):
60
50
ENTER SECOND POINT (X2, Y2):
40
10
SLOPE (m) = 2.000000
DDA Line Drawing Algorithm

```
// m = 0
```


D:\Documents\NITS\Semester VI\(LAB) CS317 Graphics and Multimedia\LAB I\L01Q...
ENTER TWO END POINTS OF THE LINE TO BE DRAWN
ENTER FIRST POINT (X1, Y1):
50
60
ENTER SECOND POINT (X2, Y2):
10
60
SLOPE (m) = -0.000000
DDA Line Drawing Algorithm

// m < 0


D:\Documents\NITS\Semester VI\(LAB) CS317 Graphics and Multimedia\LAB I\L01Q02.exe
ENTER TWO END POINTS OF THE LINE TO BE DRAWN
ENTER FIRST POINT (X1, Y1):
60
10
ENTER SECOND POINT (X2, Y2):
40
50
SLOPE (m) = -2.000000
DDA Line Drawing Algorithm

3. **Implementation of Bresenham's Line drawing algorithm. "The program should take input of the initial end-point and the final end-point, and verify whether the drawn line is smooth one or not. The program should be generic so that it works for integer values. Check for the slop of the line (i.e. m>0, m=0 and m<0)".**

➔ **CODE:**

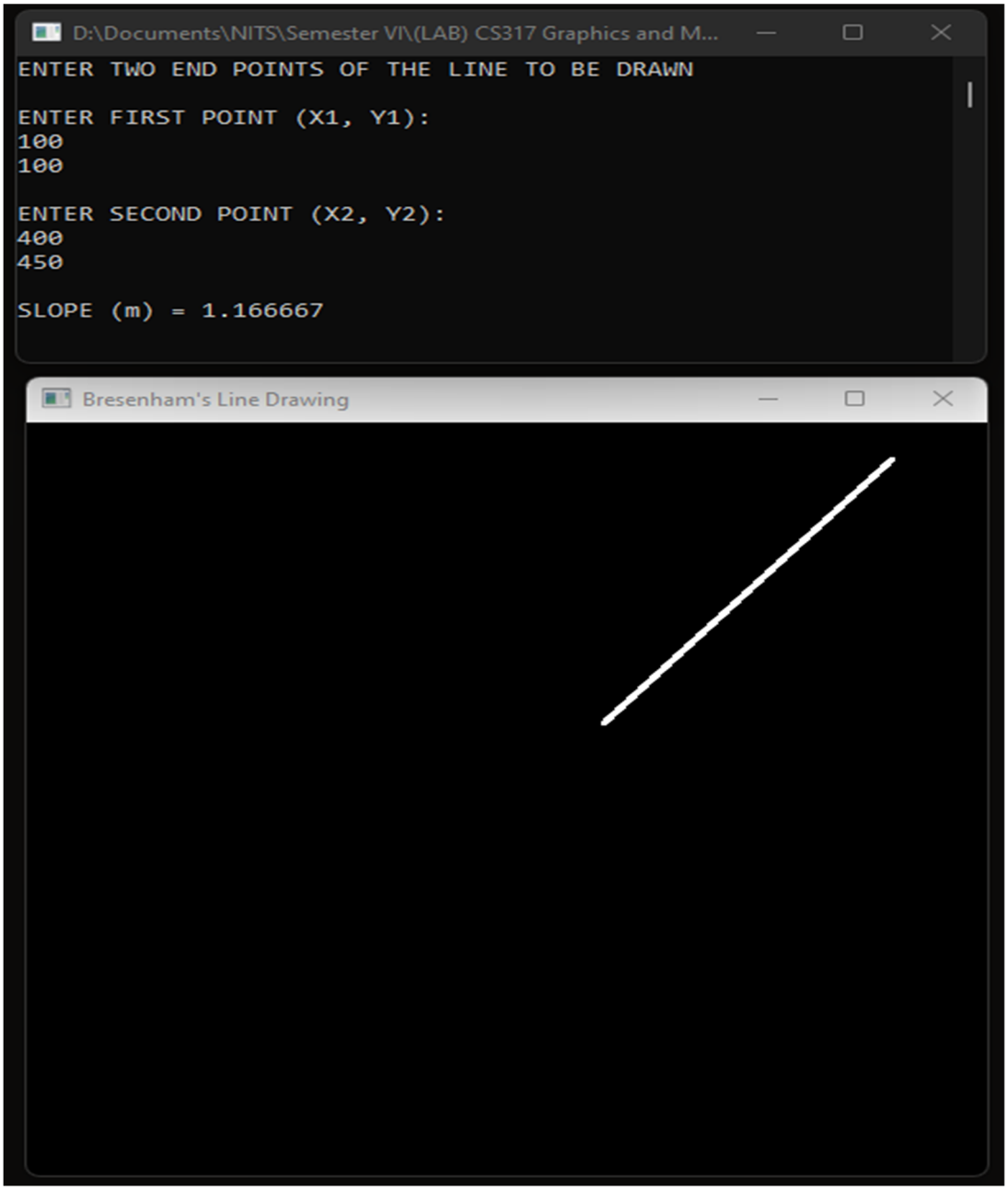
```
#include <GL/glut.h>
#include <stdio.h>

int x1, y1, x2, y2;

void myInit () {
    glClear (GL_COLOR_BUFFER_BIT);
    glClearColor (0.0, 0.0, 0.0, 0.0);
    glColor3f (1.0, 1.0, 1.0);
    glPointSize (3.0);
    glMatrixMode (GL_PROJECTION);
    glLoadIdentity ();
    gluOrtho2D (-500, 500, -500, 500);
}

void draw_pixel (int xx, int yy) {
    glBegin (GL_POINTS);
        glColor3f (1.0, 1.0, 1.0);
        glVertex2i (xx, yy);
    glEnd ();
}

void draw_line () {
    int dx, dy, ii, ee;
    int incx, incy, inc1, inc2;
    int xx, yy;
    dx = x2 - x1;
    dy = y2 - y1;
    if (dx < 0)
        dx = -dx;
    if (dy < 0)
        dy = -dy;
    incx = 1;
    if (x2 < x1)
        incx = -1;
    incy = 1;
    if (y2 < y1)
        incy = -1;
    xx = x1;
    yy = y1;
```

```
    if (dx > dy) {
        draw_pixel (xx, yy);
        ee = 2 * dy - dx;
        inc1 = 2 * (dy - dx);
        inc2 = 2 * dy;
        for (ii = 0; ii < dx; ++ii) {
            if (ee >= 0) {
                yy += incy;
                ee += inc1;
            }
            else
                ee += inc2;
            xx += incx;
            draw_pixel(xx, yy);
        }
    }
    else {
        draw_pixel (xx, yy);
        ee = 2 * dx - dy;
        inc1 = 2 * (dx - dy);
        inc2 = 2 * dx;
        for (ii = 0; ii < dy; ++ii) {
            if (ee >= 0) {
                xx += incx;
                ee += inc1;
            }
            else
                ee += inc2;
            yy += incy;
            draw_pixel (xx, yy);
        }
    }

    glFlush ();
}

int main (int argc, char **argv) {
    printf ("ENTER TWO END POINTS OF THE LINE TO BE DRAWN\n");
    printf ("\nENTER FIRST POINT (X1, Y1):\n");
    scanf (" %d %d", &x1, &y1);
    printf ("\nENTER SECOND POINT (X2, Y2):\n");
    scanf (" %d %d", &x2, &y2);
    printf ("\nSLOPE (m) = %f", ((float)y2-(float)y1)/((float)x2-
(float)x1));

    glutInit (&argc, argv);
    glutInitDisplayMode (GLUT_SINGLE|GLUT_RGB);
    glutInitWindowSize (500, 500);
```

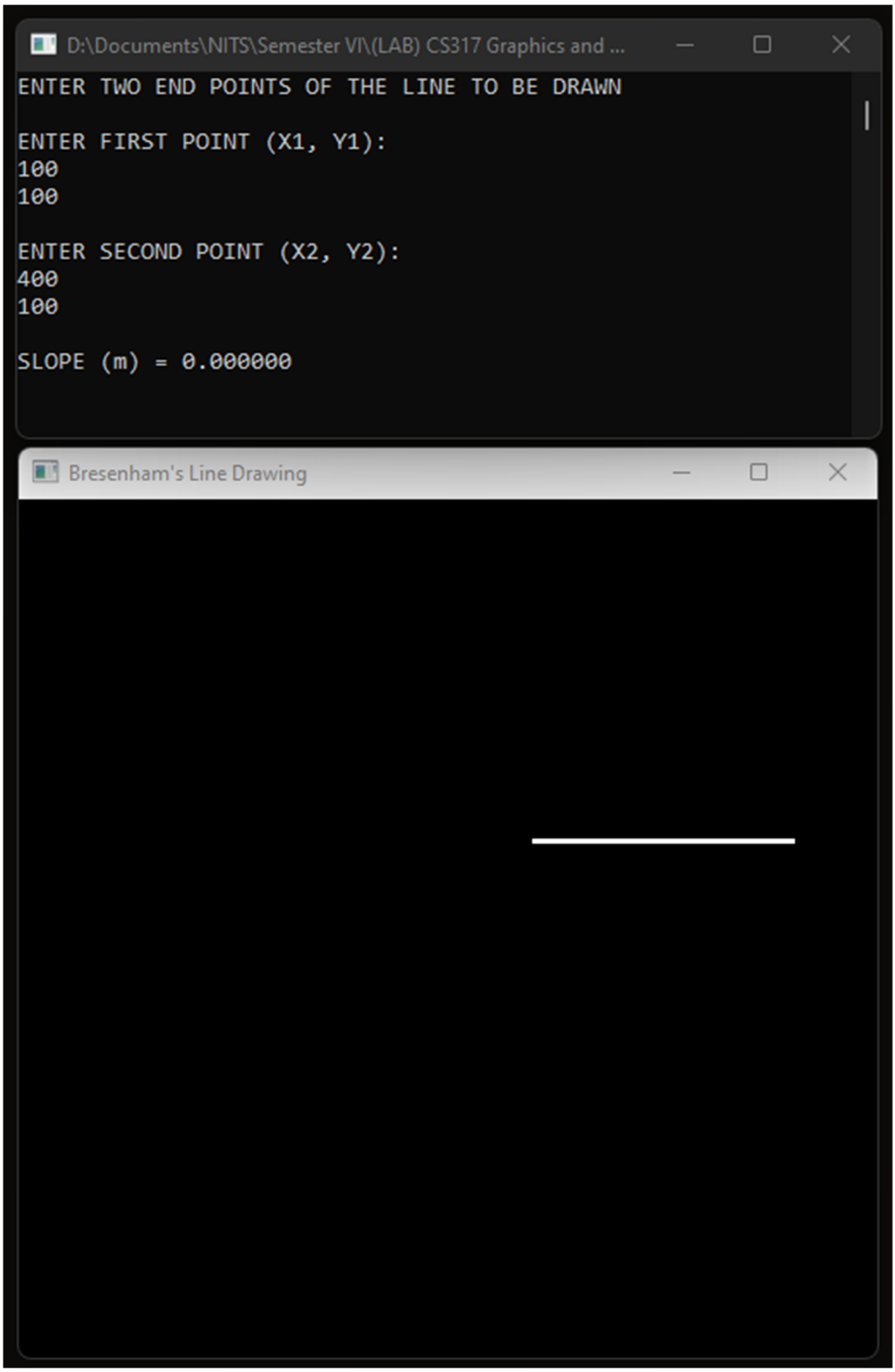
```
        glutInitWindowPosition (0, 0);
        glutCreateWindow ("Bresenham's Line Drawing");
        myInit ();
        glutDisplayFunc (draw_line);
        glutMainLoop ();

        return 0;
    }
```

OUTPUT:

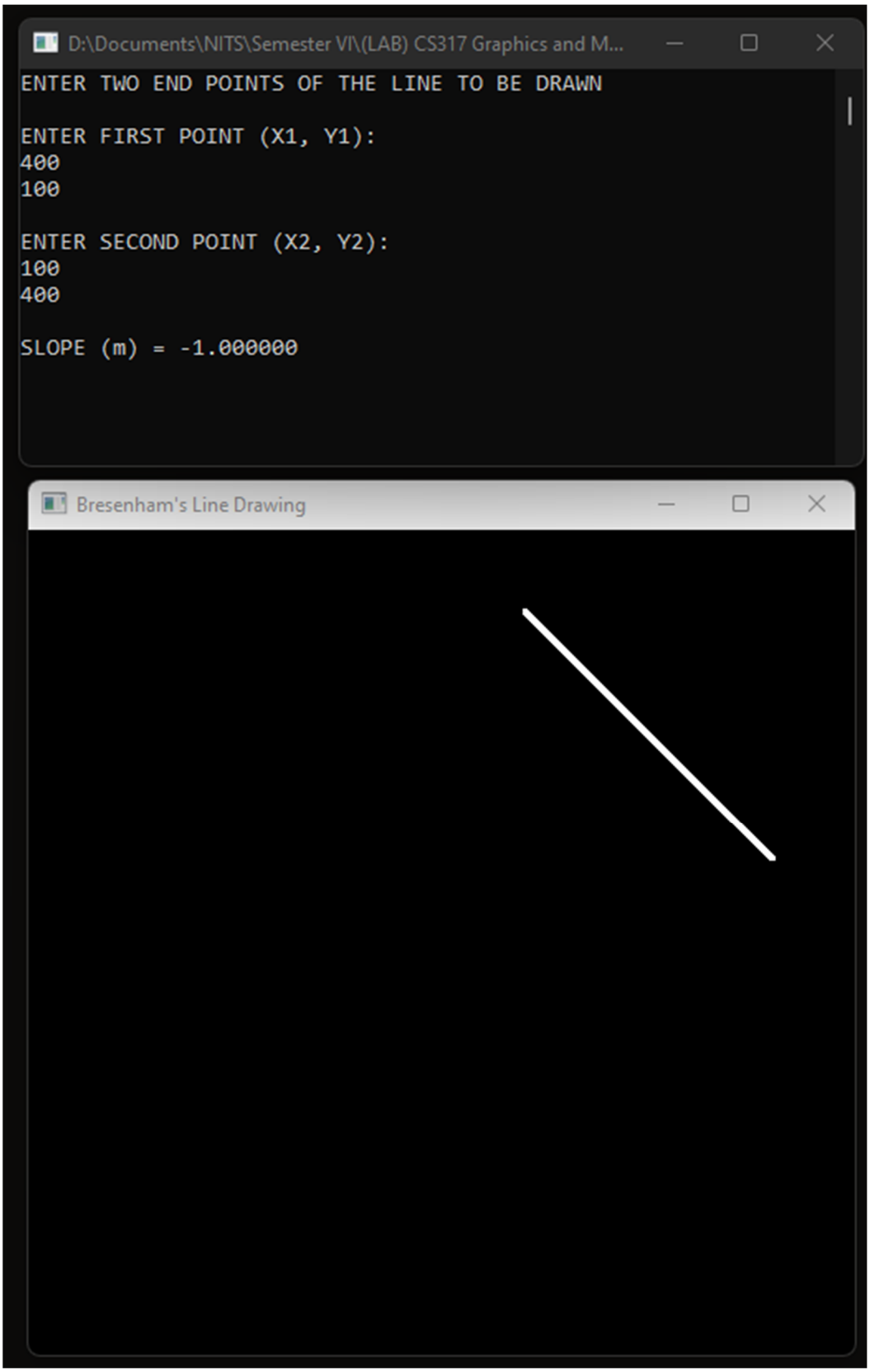
// m > 0



```
// m = 0
```


D:\Documents\NITS\Semester VI\(LAB) CS317 Graphics and ...
ENTER TWO END POINTS OF THE LINE TO BE DRAWN
ENTER FIRST POINT (X1, Y1):
100
100
ENTER SECOND POINT (X2, Y2):
400
100
SLOPE (m) = 0.000000
Bresenham's Line Drawing

// m < 0


D:\Documents\NITS\Semester VI\(LAB) CS317 Graphics and M...
ENTER TWO END POINTS OF THE LINE TO BE DRAWN
ENTER FIRST POINT (X1, Y1):
400
100
ENTER SECOND POINT (X2, Y2):
100
400
SLOPE (m) = -1.000000
Bresenham's Line Drawing