

1. Program implement Bresenham's line drawing algorithm for all types of slope.**Objective:**

In this program, students will learn to create window and to draw a line using Brenham's line drawing algorithm using openGL functions.

Program

```
// Lab1: Bresenham's Line Drawing
#include<windows.h>
#include <GL/glut.h>
#include <stdio.h>

int x1, y1, x2, y2; // line endpoints

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

void draw_pixel(int x, int y)
{
    glBegin(GL_POINTS);
    glVertex2i(x, y); //Plot Pixel
    glEnd();
}

void draw_line(int x1, int x2, int y1, int y2)
{
    {
        int dx, dy, i, e,m;
        int incx, incy, inc1, inc2;
        int x,y;

        dx = x2-x1; //Calculate constants  $\Delta x$ ,  $\Delta y$ 
        dy = y2-y1;
        m=dy/dx;
        printf("Slope m=%d",m);

        if (dx < 0) dx = -dx; //if dx is -ve change to +ve
        if (dy < 0) dy = -dy; //if dy is -ve change to +ve

        incx = 1;
        if (x2 < x1) incx = -1; //for -ve slope
        incy = 1;
        if (y2 < y1) incy = -1; //for -ve slope
        x = x1; y = y1;
```

```
if (dx > dy) //For a line with positive slope less than 1.0
{
draw_pixel(x, y); //plot starting point
e = 2 * dy-dx; //Initial decision variable  $p_0 = 2\Delta y - \Delta x$ 
inc1 = 2*(dy-dx); //Calculate constant  $2\Delta y - 2\Delta x$ 
inc2 = 2*dy; //Calculate constant  $2\Delta y$ 
for (i=0; i<dx; i++)
{
if (e >= 0) //If  $p_k \geq 0$ , thenext point to plot is  $(x_{k+1}, y_{k+1})$  and  $p_{k+1} = p_k + 2\Delta y - 2\Delta x$ 
{
y += incy;
e += inc1;
}
else //If  $p_k < 0$ , the next point to plot is  $(x_{k+1}, y_k)$  and  $p_{k+1} = p_k + 2\Delta y$ 
{
e += inc2;
}
x += incx;
draw_pixel(x, y); //plot new point
} // endfor
} //endif

else //For a line with positive slope greater than 1.0, interchange the roles of the x and y
//directions.
{
draw_pixel(x, y);
e = 2*dx-dy;
inc1 = 2*(dx-dy);
inc2 = 2*dx;
for (i=0; i<dy; i++)
{
if (e >= 0)
{
x += incx;
e += inc1;
}
else
{
e += inc2;
}
y += incy;
draw_pixel(x, y); //plot new point
} //endfor
} //endelse
} //end draw_line

void myDisplay()
{
draw_line(x1, x2, y1, y2);
glFlush();
}
```

```
int main(int argc, char **argv)
{
    printf( "Enter (x1, y1, x2, y2)\n");
    scanf("%d %d %d %d", &x1, &y1, &x2, &y2);
    glutInit(&argc, argv);
    glutInitDisplayMode(GLUT_SINGLE|GLUT_RGB);
    glutInitWindowSize(500, 500);
    glutInitWindowPosition(0, 0);
    glutCreateWindow("Bresenham's Line Drawing");
    myInit();
    glutDisplayFunc(myDisplay);
    glutMainLoop();
    return 0;
}
```

Output:

