

1. Implement Bresenham's Line drawing algorithm for all types of slope.

```
#include<GL/glut.h>
#include<stdio.h>
int x1, y1, x2, y2;

void draw_pixel(int x, int y)
{
    glColor3f(1.0,0.0,0.0);
    glBegin(GL_POINTS);
    glVertex2i(x, y);
    glEnd();
}

void bresenhams_line_draw(int x1, int y1, int x2, int y2)
{
    int dx = x2 - x1;           // x difference
    int dy = y2 - y1;           // y difference
    int m = dy/dx;              // slope

    if (m < 1)
    {
        int decision_parameter = 2*dy - dx;
        int x = x1;             // initial x
        int y = y1;             // initial y
        if (dx < 0)              // decide the first point and second point
        {
            x = x2;             // making second point as first point
            y = y2;
            x2 = x1;
        }
        draw_pixel (x, y);       // plot a point
        while (x < x2)           // from 1st point to 2nd point
        {
            if (decision_parameter >= 0)
            {
                x = x+1;
                y = y+1;
                decision_parameter = decision_parameter + 2*dy - 2*dx * (y+1 - y);
            }
            else
            {
                x = x+1;
                y = y;
                decision_parameter = decision_parameter + 2*dy - 2*dx * (y - y);
            }
            draw_pixel (x, y);
        }
    }
}
```

```

else if (m > 1)
{
    int decision_parameter = 2*dx - dy;
    int x = x1;           // initial x
    int y = y1;           // initial y
    if (dy < 0)
    {
        x = x2;
        y = y2;
        y2 = y1;
    }
    draw_pixel (x, y);
    while (y < y2)
    {
        if (decision_parameter >= 0)
        {
            x = x+1;
            y = y+1;
            decision_parameter = decision_parameter + 2*dx - 2*dy * (x+1 - x);
        }
        else
        {
            y = y+1;
            x = x;
            decision_parameter = decision_parameter + 2*dx - 2*dy * (x- x);
        }
        draw_pixel(x, y);
    }
}

else if (m == 1)
{
    int x = x1;
    int y = y1;
    draw_pixel (x, y);
    while (x < x2)
    {
        x = x+1;
        y = y+1;
        draw_pixel (x, y);
    }
}
}

```

```

void init()
{
    glClearColor(1,1,1,1);
    gluOrtho2D(0.0, 500.0, 0.0, 500.0);    // left ->0, right ->500, bottom ->0, top ->500
}

void display()
{
    glClear(GL_COLOR_BUFFER_BIT);
    bresenhams_line_draw(x1, y1, x2, y2);
    glFlush();
}

int main(int argc, char **argv)
{
    printf("Enter Start Points (x1,y1)\n");
    scanf("%d %d", &x1, &y1);                // 1st point from user

    printf("Enter End Points (x2,y2)\n");
    scanf("%d %d", &x2, &y2);                // 2nd point from user

    glutInit(&argc, argv);                    // initialize graphics system
    glutInitDisplayMode(GLUT_SINGLE|GLUT_RGB); //single buffered mode with RGB colour variants
    glutInitWindowSize(500, 500);            // 500 by 500 window size
    glutInitWindowPosition(220, 200);         // where do you wanna see your window
    glutCreateWindow("Bresenham's Line Drawing"); // the title of your window

    init();                                    // initialize the canvas

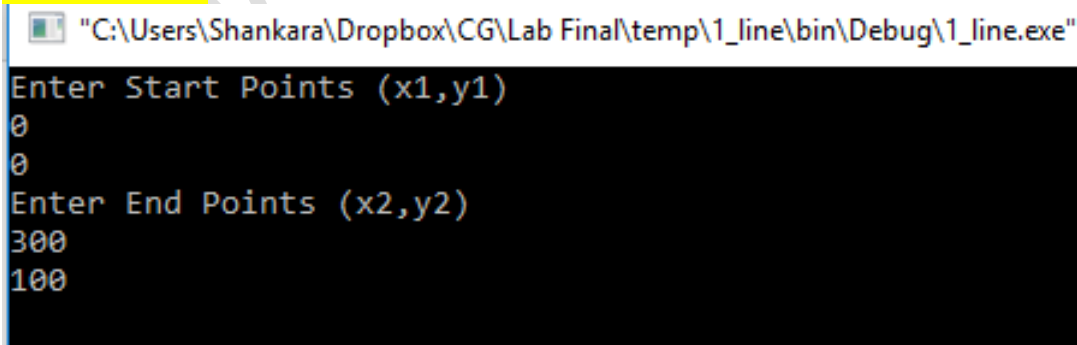
    glutDisplayFunc(display);                 // call display function

    glutMainLoop();                           // run forever
}

```

OUTPUT

Case 1: $m < 1$

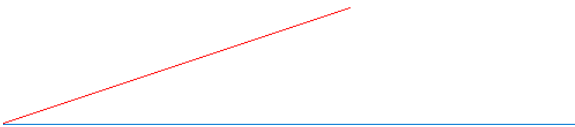
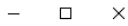


```

"C:\Users\Shankara\Dropbox\CG\Lab Final\temp\1_line\bin\Debug\1_line.exe"
Enter Start Points (x1,y1)
0
0
Enter End Points (x2,y2)
300
100

```

Bresenham's Line Drawing

**Case 2: $m > 1$**

"C:\Users\Shankara\Dropbox\CG\Lab Final\temp\1_line\bin\Debug\1_line.exe"

Enter Start Points (x1,y1)

0

0

Enter End Points (x2,y2)

50

400

Bresenham's Line Drawing



Case 3: $m = 1$

```
"C:\Users\Shankara\Dropbox\CG\Lab Final\temp\1_line\bin\Debug\1_line.exe"  
Enter Start Points (x1,y1)  
0  
0  
Enter End Points (x2,y2)  
400  
400
```

