

a) BRESENHAM'S ALGORITHM**AIM**

Write a program to draw a line using Bresenham's algorithm.

ALGORITHM

Step 1: Start

Step 2: Initialize the glut library toolkit

Step 3: Initialize window size and position

Step 4: Read x1, x2, y1, y2

Step 5:

```
glutCreateWindow("Bresenham Algorithm")
```

```
glutDisplayFunc(bresenham)
```

```
glutMainLoop()
```

Step 6: Create redrawing function bresenham()

```
def bresenham():
```

```
    Set pixel(x1, y1)
```

```
    dx, dy = abs(x2 - x1), abs(y2 - y1)
```

```
    if dx > dy:
```

```
        p = (2 * dy) - dx
```

```
        y = y1
```

```
        for x in range(x1 + 1, x2 + 1):
```

```
            if p < 0:
```

```
                p += 2 * dy
```

```
            else:
```

```
                p += (2 * dy) - (2 * dx)
```

```
                y += 1
```

```
            Set pixel(x, y)
```

```
    else:
```

```
        Perform same process for y instead of x
```

Step 7: Stop

PROGRAM

```
from OpenGL.GL import *
```

```

from OpenGL.GLU import *
from OpenGL.GLUT import *
import sys

WINDOW_SIZE = 500
SCALE = 100

x1 = y1 = 0
x2 = y2 = 25

def bresenham():
    glClear(GL_COLOR_BUFFER_BIT)
    glColor3f(1, 0, 0)
    glPointSize(5)

    glBegin(GL_POINTS)

    global x1, x2, y1, y2
    glVertex2f(x1 / SCALE, y1 / SCALE)
    dx, dy = abs(x2 - x1), abs(y2 - y1)
    if dx > dy:
        p = (2 * dy) - dx
        y = y1
        for x in range(x1 + 1, x2 + 1):
            if p < 0:
                p += 2 * dy
            else:
                p += (2 * dy) - (2 * dx)
                y += 1
            glVertex2f(x / SCALE, y / SCALE)
    else:
        p = (2 * dx) - dy
        x = x1
        for y in range(y1 + 1, y2 + 1):

```

```

if p < 0:
    p += 2 * dx
else:
    p += (2 * dx) - (2 * dy)
    x += 1
glVertex2f(x / SCALE, y / SCALE)

```

```

glEnd()
glFlush()

```

```

def main():
    glutInit(sys.argv)
    glutInitDisplayMode(GLUT_SINGLE | GLUT_RGB)
    glutInitWindowSize(WINDOW_SIZE, WINDOW_SIZE)
    glutInitWindowPosition(50, 50)

```

```

global x1, x2, y1, y2
x1 = int(input("Enter x coordinate of first endpoint "))
y1 = int(input("Enter y coordinate of first endpoint "))
x2 = int(input("Enter x coordinate of second endpoint "))
y2 = int(input("Enter y coordinate of second endpoint "))
if x1 > x2:
    x1, x2 = x2, x1
    y1, y2 = y2, y1
glutCreateWindow("Bresenham Algorithm")
glutDisplayFunc(bresenham)

```

```

glutMainLoop()

```

```

main()

```

RESULT

The required program has been created.

INPUT/OUTPUT

```
Enter x coordinate of first endpoint 1  
Enter y coordinate of first endpoint 2  
Enter x coordinate of second endpoint 10  
Enter y coordinate of second endpoint 15
```

Bresenham Algorithm

—

□

×

