**LAB-3**

**POINTER and NAMESPACE**

**Objective: -**

* Determine intermediate points to approximate a straight line on a digital screen through incremental step-by-step calculations.
* Produce lines with minimal error, efficiently managing both steep and shallow slopes.

**Theory: -**

The Digital Differential Analyzer (DDA) algorithm is a popular method in computer graphics for creating straight lines. It incrementally computes intermediate points between the start and end coordinates, ensuring accurate and smooth rasterization. DDA is versatile, efficient for varying slopes, and simplifies line rendering on digital displays or screens.

**Algorithm: -**

1. Input the starting and ending coordinates of the line

2. Calculate the differences:

3. Determine the number of steps:

4. Calculate the increment for each step:

5. Initialize the starting point:

6. Iterate through the number of steps:

- Plot the point

- Increment:

7. Stop once all steps are plotted.

**Code: -**

import matplotlib.pyplot as plt

def dda():

x1 = int(input('Enter the value of x1: '))

y1 = int(input('Enter the value of y1: '))

x2 = int(input('Enter the value of x2: '))

y2 = int(input('Enter the value of y2: '))

dx = x2 - x1

dy = y2 - y1

steps = max(abs(dx), abs(dy))

xes = []

yes = []

x, y = x1, y1

for i in range(steps):

xes.append(x)

yes.append(y)

xi = dx / steps

yi = dy / steps

x = x + xi

y = y + yi

plt.plot(xes, yes, marker='x')

plt.show()

dda()

**Output: -**

Enter the value of x1: 78

Enter the value of y1: 34

Enter the value of x2: 12

Enter the value of y2: 67

**A graph with a line

Description automatically generated with medium confidence**

**Discussion: -**

The Digital Differential Analyzer (DDA) algorithm is a fundamental technique in computer graphics for rasterizing lines. Its incremental approach ensures efficiency and accuracy, suitable for diverse slopes. DDA's simplicity makes it ideal for teaching and implementation in Python. However, it requires floating-point calculations, which can sometimes introduce rounding errors on low-performance systems.

**Conclusion: -**

In conclusion, the Digital Differential Analyzer (DDA) algorithm is an effective and straightforward method for line generation in computer graphics. Its incremental approach ensures precise rendering of lines by calculating intermediate points efficiently, making it ideal for digital raster displays. Implementing DDA in Python highlights its simplicity and adaptability to various slopes, offering a foundational technique for graphics programming. This algorithm remains a fundamental tool for understanding line rasterization in modern computer graphics applications.