# Lab Report 2

## Title: To use DDA algorithm to draw a line between given points.

## Theory:

In any 2-Dimensional plane if we connect two points (x1, y1) and (x2, y2), we get a line segment. But in the case of computer graphics, we can not directly join any two coordinate points, for that we should calculate intermediate points’ coordinates and put a pixel for each intermediate point, of the desired color with help of functions like putpixel(x, y, K) in C, where (x,y) is our co-ordinate and K denotes some color.

For using graphics functions, our system output screen is treated as a coordinate system where the coordinate of the top-left corner is (0, 0) and as we move down our y-ordinate increases and as we move right our x-ordinate increases for any point (x, y).

Now, for generating any line segment we need intermediate points and for calculating them we can use a basic algorithm called DDA(Digital differential analyzer) line generating algorithm.

**DDA Algorithm:**

1. Start
2. Take the coordinates of initial point (x1,y1) and the final point (x2,y2).
3. Find the difference between the points by using the formula:   
    dx = x2-x1;  
    dy = y2-y1;
4. Compare the absolute value of the differences to get the number of steps.  
    If abs(dx) > abs(dy)  
    steps = dx;  
    Else   
    steps = dy;
5. Calculate the increment of x1 and y1 in each step.  
    dx = dx/steps;  
    dy = dy/steps;
6. Put pixel in (x1,y1).
7. Repeat step 5 while adding dx to x and dy to y until steps reaches zero.  
    while (steps != 0)  
    putpixel(x1,y1,K);  
    x1 = x1+dx;  
    y1 = y1+dy;  
    end while
8. Stop

## **The code:**

The code for the asked program is given below:

// DDA algorithm to draw a line.

// Including preprocessor directives.

**#**include <stdio.h>

**#**include <graphics.h>

**#**include <math.h>

// Declaring main function.

int main(int argc, char const \*argv[])

{

// Declaring variables.

int x1,y1,x2,y2,dx,dy,steps;

// Asking for initial point.

printf("Enter the initial point\n");

scanf("%d %d",&x1,&y1);

// Asking for final point.

printf("Enter the final point\n");

scanf("%d %d",&x2,&y2);

// Calculating the differences between initial and final points.

dx=abs(x2-x1);

dy=abs(y2-y1);

// Determining the number of steps.

if (dx>=dy)

{

steps=dx;

}

else

{

steps=dy;

}

// Determining the increment per step.

dy/=steps;

dx/=steps;

// Initializing graphics variables.

int gd=DETECT,gm;

// Initializing graphics.

initgraph(&gd,&gm,NULL);

// Drawing individual pixels in a loop.

for (int i = 0; i < steps; i++)

{

putpixel(x1,y1,5);

x1+=dx;

y1+=dy;

delay(100);

}

// Concluding the program.

delay(500000);

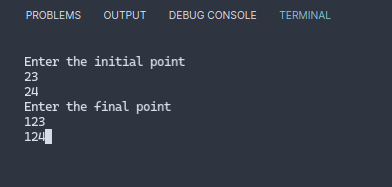
closegraph();

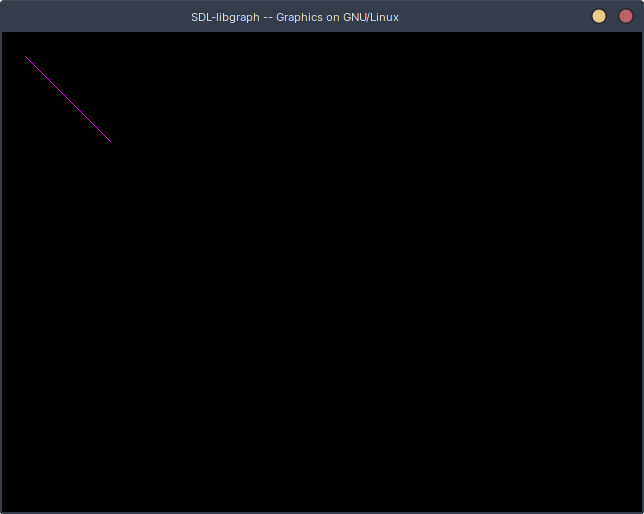
return 0;

}

## **The output:**

The output of the asked program is given below:

Figure 1: Input the initial and final points of the line

Figure 2: Line drawn using DDA algorithm

## **Conclusion:**

Thus, as shown in the program above, we can draw a line by drawing individual pixels with the Digital Differential Analyzer (DDA) algorithm using the various functions available in the graphics.h header file.