# 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. Calculate slope between the points by using the formula:   
    Slope=(y2-y1)/(x2-x1);
4. While(x1!=x2 && y1!=y2)

putpixel(x1,y1,K);

if slope > 1  
 x1=x1+(1/slope);

y1=y1+1;

else if (slope<1)

x1=x1+1;

y1=y1+(1/slope);

else

x1=x1+1;

y1=y1+1;

end while

1. 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[])

{

//initializing graphics

int gd=DETECT,gm;

initgraph(&gd,&gm,NULL);

//declaring variables

float x1,x2,y1,y2,slope;

int count;

//Taking co-ordinates as input

cout<<"Enter values of (x1,y1):\n";

cin>>x1>>y1;

cout<<"Enter values of (x2,y2):\n";

cin>>x2>>y2;

//calculating slope

slope=(y2-y1)/(x2-x1);

//checking condition

while(x1!=x2 && y1!=y2){

//plotting pixels on console

putpixel(x1,y1,WHITE);

if (slope>1){

x1=x1+(1/slope);

y1=y1+1;

}

else if(slope<1){

x1=x1+1;

y1=y1+(1/slope);

}

else{

x1=x1+1;

y1=y1+1;

}

}

//Displaying my roll number

outtextxy(400,100, "078BCT028");

//terminating program

delay(50000);

closegraph();

return 0;

}

## The output:

The output of the asked program is given below:

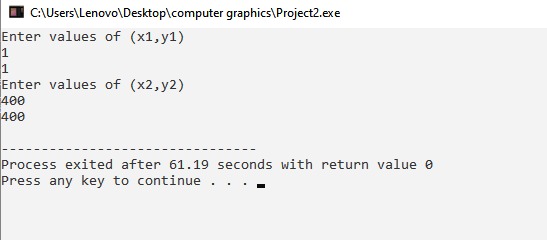
Figure 1: Input the initial and final points of the line

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.