* Write a program to implement DDA Line Drawing Algorithm.

**Algorithm:-**

Digital Differential Analyzer algorithm is also known as an **incremental method**of scan conversion. In this algorithm, we can perform the calculation in a step by step manner. We use the previous step result in the next step.

As we know the general equation of the straight line is:

**y = mx + c**

/ calculate dx , dy

dx = X1 - X0;

dy = Y1 - Y0;

// Depending upon absolute value of dx & dy

// choose number of steps to put pixel as

// steps = abs(dx) > abs(dy) ? abs(dx) : abs(dy)

steps = abs(dx) > abs(dy) ? abs(dx) : abs(dy);

// calculate increment in x & y for each steps

Xinc = dx / (float) steps;

Yinc = dy / (float) steps;

// Put pixel for each step

X = X0;

Y = Y0;

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

{

putpixel (X,Y,WHITE);

X += Xinc;

Y += Yinc;

}

.

**Program:-**

**// C program for DDA line generation**

**#include<stdio.h>**

**#include<graphics.h>**

**//Function for finding absolute value**

**int abs (int n)**

**{**

**return ( (n>0) ? n : ( n \* (-1)));**

**}**

**//DDA Function for line generation**

**void DDA(int X0, int Y0, int X1, int Y1)**

**{**

**// calculate dx & dy**

**int dx = X1 - X0;**

**int dy = Y1 - Y0;**

**// calculate steps required for generating pixels**

**int steps = abs(dx) > abs(dy) ? abs(dx) : abs(dy);**

**// calculate increment in x & y for each steps**

**float Xinc = dx / (float) steps;**

**float Yinc = dy / (float) steps;**

**// Put pixel for each step**

**float X = X0;**

**float Y = Y0;**

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

**{**

**putpixel (X,Y,RED); // put pixel at (X,Y)**

**X += Xinc; // increment in x at each step**

**Y += Yinc; // increment in y at each step**

**delay(100); // for visualization of line-**

**// generation step by step**

**}**

**}**

**// Driver program**

**int main()**

**{**

**int gd = DETECT, gm;**

**// Initialize graphics function**

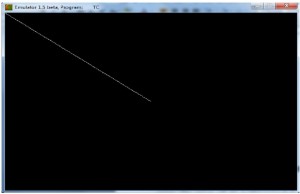
**initgraph (&gd, &gm, "");**

**int X0 = 2, Y0 = 2, X1 = 14, Y1 = 16;**

**DDA(2, 2, 14, 16);**

**return 0;**

**}**

****

**Figure:- Output of the program**