**Aim:** To implement DDA algorithms for drawing a line segment between two given end points.

**Objective:** Draw the line using (vector) generation algorithms which determine the pixels that should be turned ON are called as digital differential analyzer (DDA).It is one of the techniques for obtaining a rasterized straight line. This algorithm can be used to draw the line in all the quadrants.

**Theory:**

DDA algorithm is an incremental scan conversion method. Here we perform calculations at each step using the results from the preceding step. The characteristic of the DDA algorithm is to take unit steps along one coordinate and compute the corresponding values along the other coordinate. Digital Differential Analyzer (DDA) algorithm is the simple line generation algorithm which is explained step by step here.

**Algorithm:**

**{**

**dx=x2-x1;**

**dy=y2-y1;**

**if(abs(dx) > abs(dy)) {**

**step = dx;**

**} else {**

**step = dy;**

**}**

**Xn = dx/step;**

**Yn = dy/step;**

**for(i=0;i<=step;i++)**

**{**

**putpixel(x1,y1,RED);**

**x1 = x1 + Xn;**

**y1 = y1 + Yn;**

**}**

**}**

**Program:**

#include<stdio.h>

#include<conio.h>

#include<graphics.h>

#include<math.h>

void main()

{

int gd=DETECT,gm;

int i;

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

initgraph(&gd,&gm,"C:\\turboc3\\bgi");

printf("Enter values of x1 and x2:");

scanf("%d %d",&x1,&x2);

printf("Enter values of y1 and y2:");

scanf("%d %d",&y1,&y2);

dx=x2-x1;

dy=y2-y1;

if(abs(dx)>abs(dy))

steps=abs(dx);

else steps=abs(dy);

xinc=dx/steps;

yinc=dy/steps;

for(i=1;i<=steps;i++)

{

putpixel(x1,x2,RED);

x1=x1+xinc;

y1=y1+yinc;

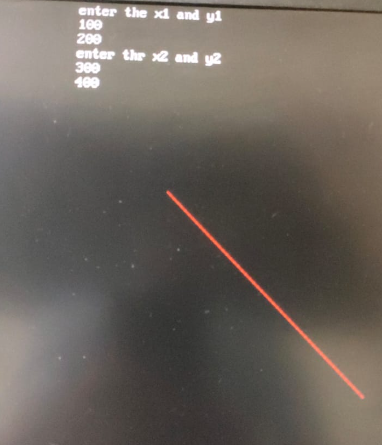
}

getch();

closegraph();

}

**Output:**



**Conclusion:**

Comment on -

1) Pixel - A pixel is the smallest unit of a digital image or display, representing a single point in a two-dimensional grid. Pixels are the building blocks of digital visuals, and they play a crucial role in rendering images, text, and graphics on screens. Each pixel contains information about color and brightness, collectively forming the images we see on screens.

2) Equation for line- The equation for a straight line, y = mx + b, is fundamental in mathematics and computer graphics. It represents a linear relationship between the x and y coordinates of points on the line. The slope (m) determines the line's steepness, while the y-intercept (b) specifies the point where the line crosses the y-axis. This equation is essential for describing and rendering lines in graphics.

3)Need of line drawing algorithm- Line drawing algorithms are essential in computer graphics and image processing for several reasons:

* Efficiency: Algorithms like Bresenham's provide an efficient way to draw lines on a discrete grid of pixels, optimizing computational resources.
* Accuracy: They produce precise results, ensuring that the line closely matches the specified endpoints.
* Graphics Rendering: Line drawing is a fundamental component of rendering graphics, including shapes, curves, and complex images.
* Geometric Computations: Line drawing algorithms are foundational for geometric computations, making them useful in CAD (Computer-Aided Design), gaming, and simulations.

4)Slow or fast- The efficiency of a line drawing algorithm can vary significantly. Bresenham's Line Drawing Algorithm, in particular, is known for its speed and efficiency. It uses integer arithmetic and takes advantage of the symmetry of lines to minimize calculations, making it one of the fastest algorithms for drawing lines on a grid of pixels. Its speed is especially beneficial for real-time graphics applications like video games and animations.