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

**Objective:**

Draw a line using Bresenham's line algorithm that determines the points of an n-dimensional raster that should be selected to form a close approximation to a straight line between two points

**Theory:**

In Bresenham’s line algorithm pixel positions along the line path are obtained by determining the pixels i.e. nearer the line path at each step.

**Algorithm –**

x=x1;

y=y1;

dx=x2-x1;

dy=y2-y1;

p=2dy-dx;

while(x<=x2)

{

putpixel(x,y,RED);

x++;

if(p<0)

{

p=p+2dy;

}

else

{

p=p+2dy-2dx;

y++;

}

}

**Program -**

#include<stdio.h>

#include<conio.h>

#include<graphics.h>

#include<math.h>

void main()

{

int x1,y1,x2,y2,dx,dy,p,x,y;

int gd=DETECT,gm;

initgraph(&gd,&gm,"C:\\TURBOC3\\BGI");

clrscr();

printf("ENTER X1 AND Y1: ");

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

printf("ENTER X2 AND Y2: ");

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

x=x1;

y=y1;

dx=x2-x1;

dy=y2-y1;

p=2\*dy-dx;

while(x<=x2)

{

putpixel(x,y,RED);

x++;

if(p<0)

{

p=p+2\*dy;

}

else

{

p=p+2\*dy-2\*dx;

y++;

}

}

getch();

closegraph();

}

**Output –**



**Conclusion:** Comment on -

1. **Pixel**:-In computer graphics, the display or screen is composed of a grid of individual picture elements, commonly known as pixels. Each pixel can be independently controlled to display a specific color, creating the overall image.
2. **Equation for line**:- ((y - y1) = m(x - x1)) or (y = mx + b)
3. **Need of line drawing algorithm**:-Line drawing algorithms are essential in computer graphics to render lines between specified points. Without a line drawing algorithm, displaying a straight line on a pixel grid would require individually setting each pixel position along the line, which is impractical and inefficient. The DDA algorithm provides a simple and efficient way to approximate lines on the pixel grid.
4. **Slow or fast** :-The DDA algorithm is relatively fast and suitable for real-time rendering on most systems.