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:

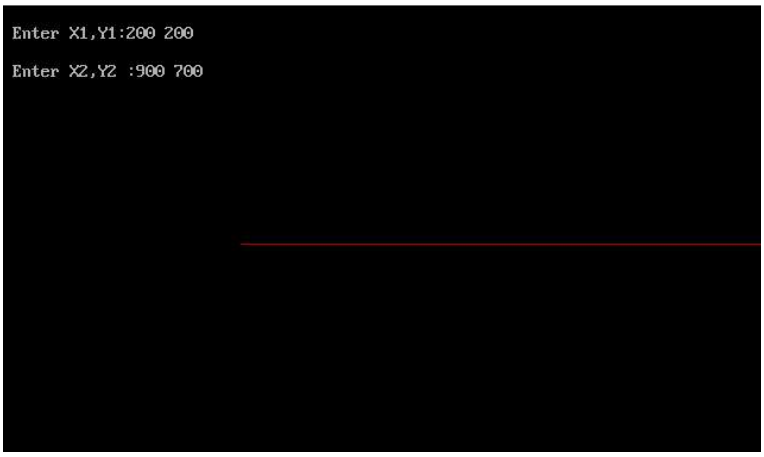
```
{
dx=x2-x1;
dy=y2-y1;
if(abs(dx)>abs(dy))
{
step=dx;
}
else
dy=y2-y1;
if(abs(dx)>abs(dy))
{
step=dx;
}
else
{
step=dy;
}
xn=dx/step;
yn=dy/step;
for(i=0;i<=step;i++)
{
putpixel(x1,y1,RED);
x1=x1+xn;
y1=y1+yn;
}
```

}

Program:

```
#include<stdio.h>
#include<conio.h>
#include<graphics.h>
#include<math.h>
void main()
{
int x1,x2,y1,y2,xn,yn,dx,dy,step,i;
int gd=DETECT,gm;
initgraph(&gd,&gm,"(:\\TURBOC3\\BGI");
printf("\n Enter x1 and y1");
scanf("%d,%d",&x1,&y1);
printf("\n Enter x2 and y2");
scanf("%d,%d",&x2,&y2);
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;
}
getch();
closegraph();
}
```

Output:



Conclusion: Comment on -

1. Pixel-The term "pixel" is short for "picture element." It is the smallest unit of an image or graphics displayed on a digital screen.
2. Equation for line-y=mx+c
3. Need of line drawing algorithm-1)Simplicity 2)Floating-point precision 3)Hardware acceleration 4)Incremental approach 5)Line thickness 6)Interpolation
4. Slow or fast-Slow

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=2*dy-dx;
while(x1<=x2)
{
 putpixel(x1,y1,RED);
 x1++;
 if(p<0)
{
 p=p+2*dy;
 }
else
{
p=p+2*dy-2*dx;
y1++;
}
}
```

Program -

```
#include<stdio.h>
#include<conio.h>
#include<graphics.h>
#include<math.h>
#include<dos.h>
```

```
void main()
{
 clrscr();
 int gd=DETECT, gm;
 int x1,y1,x2,y2,dx,dy,p;
  initgraph(&gd,&gm,"C:\\TURBOC3\\BGI");
 printf("Enter x1,y1s:\n");
 scanf("%d %d",&x1,&y1);
 printf("Enter x2,y2:\n");
 scanf("%d %d",&x2,&y2);
 dx=x2-x1;
 dy=y2-y1;
 p=2*dy-dx;
while(x1<=x2){
 putpixel(x1,y1,50);
 x1++;
 if(p<0){
 p=p+2*dy;
 }
else
{
p=p+2*dy-2*dx;
y1++;
}
}
getch();
closegraph();
}
```

Output –

Conclusion: Comment on -

1. Pixel-The term "pixel" is short for "picture element." It is the smallest unit of an image or graphics displayed on a digital screen.
2. Equation for line -y=mx+c
3. Need of line drawing algorithm-1)Efficiency2)Straight-line drawing3)Rasterization4)Hardware implementation5)Straight line approximation6)Learning and understanding