**LAB 1**

**OBJECTIVE:**

TO IMPLEMENT DDA ALGORITHM FOR DRAWING A LINE SEGMENT BETWEEN TWO GIVEN END POINTS A(x1,y1) AND B(x2,y2).

**THEORY:**

DDA stands for Digital Differential Analyzer. In the DDA algorithm, we have to approximate the value of each pixel in whole number if any value of X-coordinate or Y-coordinate is in the fractional form by the addition or subtraction of the slope(m). Here we approximate the value by round off.

In DDA we need to consider two cases; One is slope of the line less than or equal to one (|m| ≤1) and slope of the line greater than one (m| > 1).

* When |m| ≤ 1 means y2-y1 = x2-x1 or y2-y1< x2-x1 and therefore we assume x to be the major axis. Here we sample x axis at unit intervals and find the y values corresponding to each x value. We have the slope equation as

∆ y = m ∆ x  
y2-y1 = m (x2-x1)   
so, xk+1= xk+1 and yk+1= yk+m

* When |m| > 1 means y2-y1 > x2-x1 and therefore we assume y to be the major axis. Here we sample y axis at unit intervals and find the x values corresponding to each y value. We have the slope equation as

∆ y = m ∆ x  
 y2-y1 = m (x2-x1)   
so, yk+1= yk+1 and xk+1= xk+1/m

The algorithm for DDA is given in following steps:

Step1: Start

Step2: Read two end points P1(x1, y1) and P2 (x2,y2)

Step3: Calculate dx and dy using

dx = x2 – x1

dy = y2 – y1

Step4: Calculate slope m = dy/dx

Step5: if abs(m)<1

y = y1   
 for (x = x1; x<=x2; x++)   
 y = y+m

plot the pixel position with specified color: setPixel(x, round(y))

else

if abs(m>1)

for (y=y1; y<=y2; y++)

x = x+1/m

plot the pixel position with specified color: setPixel(round(x), y)

else

for(x=x1;x<=x2;x++);

x = x+1;

y = y+1;

plot the pixel position with specified color: putpixel(round(x),round(y))

Step6: Close the graph and Stop.

SOURCE CODE:

#include <stdio.h>

#include <graphics.h>

#include <math.h>

#include <conio.h>

int roundNow(float x){

if((x-(int)x)>=0.5) return (int)x+1;

else return (int)x;

}

void drawLine(int a, int b, int c, int d){

float x=a,y=b;

int dx=(c-a), dy=(d-b);

float m=dy/(float)dx;

if(fabs(m)<1){

while(x<=c){

putpixel(roundNow(x),roundNow(y),RED);

printf("%d, %d\n", roundNow(x), roundNow(y));

x+=1;

y+=m;

}

} else if(fabs(m)>1) {

while(y<=d){

putpixel(roundNow(x),roundNow(y),RED);

printf("%d, %d\n", roundNow(x), roundNow(y));

x+=1/m;

y+=1;

}

} else {

while(y<=d || x<=c){

putpixel(roundNow(x),roundNow(y),RED);

//printf("%d, %d\n", roundNow(x), roundNow(y));

x+=1;

y+=1;

}

}

}

int main()

{

int graphicsDriver=DETECT, graphicsMode;

int a,b,c,d;

//clrscr();

printf("Enter starting points: ");

scanf("%d %d", &a,&b);

printf("Enter ending points: ");

scanf("%d %d", &c,&d);

initgraph(&graphicsDriver, &graphicsMode, "C:\\TURBOC3\\BGI");

drawLine(a,b,c,d);

getch();

closegraph();

return 0x1337;

}

OUTPUT: