

**KIET Group of Institutions, Ghaziabad**

**Department of Computer Applications (NBA Accredited)**

**(An ISO – 9001: 2015 Certified & ‘A’ Grade accredited Institution by NAAC)**

**KIET Group of Institutions**

Write a program to implement DDA Line Drawing Algorithm.

ALGORITHM:

**Step1:** Start Algorithm

**Step2:** Declare x1,y1,x2,y2,dx,dy,x,y as integer variables.

**Step3:** Enter value of x1,y1,x2,y2.

**Step4:** Calculate dx = x2-x1

**Step5:** Calculate dy = y2-y1

**Step6:** If ABS (dx) > ABS (dy)  
            Then step = abs (dx)  
            Else

**Step7:** xinc=dx/step  
            yinc=dy/step  
            assign x = x1  
            assign y = y1

**Step8:** Set pixel (x, y)

**Step9:** x = x + xinc  
            y = y + yinc  
            Set pixels (Round (x), Round (y))

**Step10:** Repeat step 9 until x = x2

**Step11:** End Algorithm

C PROGRAM:

#include<graphics.h>

#include<conio.h>

#include<stdio.h>

void main()

{

int gd = DETECT ,gm, i;

float x, y,dx,dy,steps;

int x0, x1, y0, y1;

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

printf("The initial points:");

scanf("%d %d",&x0,&y0);

printf("The final points:");

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

dx = (float)(x1 - x0);

dy = (float)(y1 - y0);

if(dx>=dy)

{

steps = dx;

}

else

{

steps = dy;

}

dx = dx/steps;

dy = dy/steps;

x = x0;

y = y0;

i = 1;

while(i<= steps)

{

putpixel(x, y, RED);

x += dx;

y += dy;

i=i+1;

delay(100); // this method is used to put some delay while creating line

}

getch();

closegraph();

}

PROGRAM OUTPUT:

