Aim->Write a program to implement DDA Line Drawing Algorithm

**DDA 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 for DDA line generation

#include<stdio.h>

#include<graphics.h>

//Function for finding absolute value

int abs (int n)

{

    return ( (n>0) ? n : ( n \* (-1)));

}

//DDA Function for line generation

void DDA(int X0, int Y0, int X1, int Y1)

{

    // calculate dx & dy

    int dx = X1 - X0;

    int dy = Y1 - Y0;

    // calculate steps required for generating pixels

    int steps = abs(dx) > abs(dy) ? abs(dx) : abs(dy);

    // calculate increment in x & y for each steps

    float Xinc = dx / (float) steps;

    float Yinc = dy / (float) steps;

    // Put pixel for each step

    float X = X0;

    float Y = Y0;

    for (int i = 0; i <= steps; i++)

    {

        putpixel (X,Y,RED);  // put pixel at (X,Y)

        X += Xinc;

        Y += Yinc;

        delay(100);

     }

}

int main()

{

    int gd = DETECT, gm;

    initgraph (&gd, &gm, "");

    DDA(2, 2, 14, 16);

    return 0;

}

