**Write a program to implement DDA Line Drawing Algorithm**

**DDA Algorithm :**  
Consider one point of the line as (X0,Y0) and the second point of the line as (X1,Y1).

// calculate dx , dy

dx = X1 - X0;

dy = Y1 - Y0;

// Depending upon absolute value of dx & dy

// choose number of steps to put pixel as

// steps = abs(dx) > abs(dy) ? abs(dx) : abs(dy)

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

// calculate increment in x & y for each steps

Xinc = dx / (float) steps;

Yinc = dy / (float) steps;

// Put pixel for each step

X = X0;

Y = Y0;

for (int i = 0; i <= steps; i++)

{

putpixel (X,Y,WHITE);

X += Xinc;

Y += Yinc;

}

**PROGRAM:**

#include<iostream.h>

#include<graphics.h>

#include<math.h>

void main()

{

int gd=DETECT,gm;

initgraph(&gd,&gm,””);

int x1,x2,y1,y2,dx,dy,Xinc,Yinc,steps;

cout<<”enter values of x1 and y1”<<endl;

cin>>x1>>y1;

cout<<”enter values of x2 and y2”<<endl;

cin>>x2>>y2;

dx=x2-x1;

dy=y2-y1;

if(abs(dx)>abs(dy))

steps=abs(dx);

else

steps=abs(dy);

Xinc=dx/steps;

Yinc=dy/steps;

for(int i=0;i<steps;i++)

{

putpixel(x1,y1,RED);

x1=x1+Xinc;

y1=y1+Yinc;

delay(500);

}

}

**OUTPUT:**

****