DDA Algorithm

Write a C program to draw a line using DDA algorithm.

External Practical Experiment

COMPUTER GRAPICS AND MULTIMEDIA

Syeda Reeha Quasar

14114802719

Group – C6

# **EXTERNAL EXPERIMENT**

## **AIM:**

Write a C program to draw a line using DDA algorithm.

## **THEORY:**

DDA stands for Digital Differential Analyzer. It is an incremental method of scan conversion of line. In this method calculation is performed at each step but by using results of previous steps.

### **Advantage:**

1. It is a faster method than method of using direct use of line equation.
2. This method does not use multiplication theorem.
3. It allows us to detect the change in the value of x and y, so plotting of same point twice is not possible.
4. This method gives overflow indication when a point is repositioned.
5. It is an easy method because each step involves just two additions.

### **Disadvantage:**

1. It involves floating point additions rounding off is done. Accumulations of round off error cause accumulation of error.
2. Rounding off operations and floating-point operations consumes a lot of time.
3. It is more suitable for generating line using the software. But it is less suited for hardware implementation.

## **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

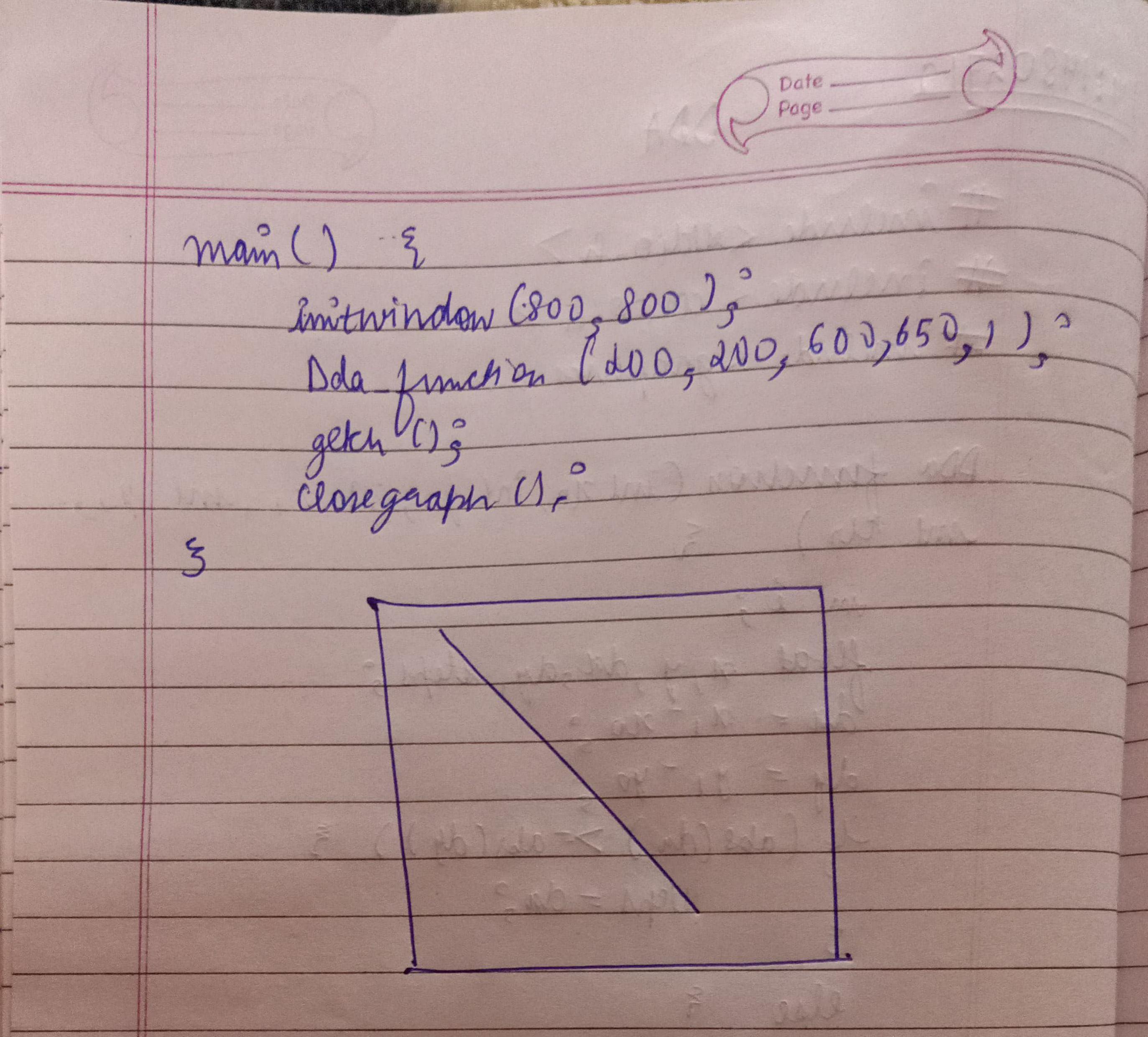
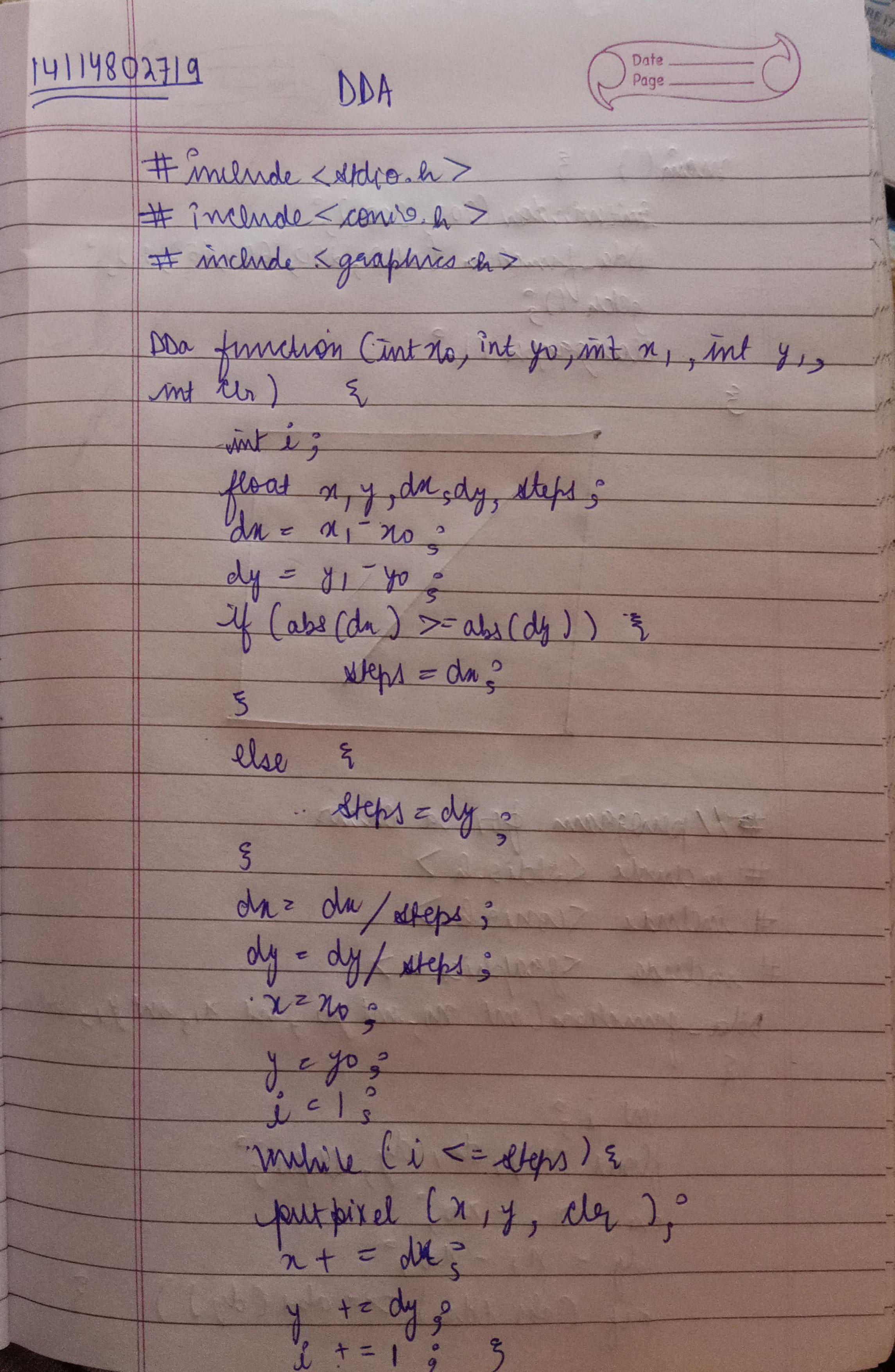
**Step7:** xinc=dx/step  
            yinc=dy/step  
            assign x = x1  
            assign y = y1

**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



# Drawing line using DDA Algorithm :

## CODE

#include<stdio.h>

#include<conio.h>

#include<graphics.h>

Dda\_function(int x0,int y0,int x1,int y1,int clr)

{

int i;

float x,y,dx,dy ,steps;

dx=x1-x0;

dy=y1-y0;

if(abs(dx)>=abs(dy)) {

steps=dx;

}

Else {

steps=dy;

}

dx=dx/steps;

dy=dy/steps;

x=x0;

y=y0;

i=1;

while(i<=steps) {

putpixel(x,y,clr);

x+=dx;

y+=dy;

i=i+1;

}

}

main(){

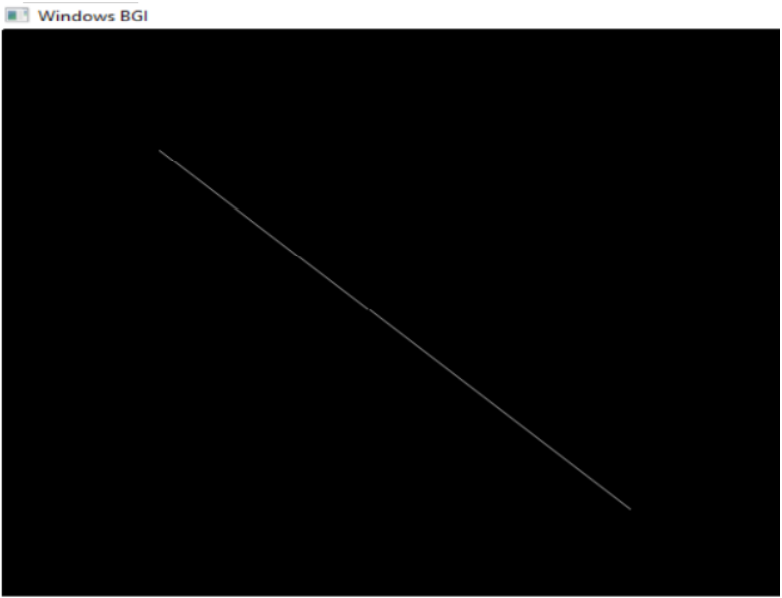
initwindow(800,800);

Dda\_function(200, 200, 600, 650, 1);

getch();

closegraph();

}



# **Source Code:**

// C program for drawing a chair

#include<stdio.h>

#include<conio.h>

#include<graphics.h>

Dda\_function(int x0,int y0,int x1,int y1,int clr)

{

int i;

float x,y,dx,dy ,steps;

dx=x1-x0;

dy=y1-y0;

if(abs(dx)>=abs(dy)){

steps=dx;

}

else{

steps=dy;

}

dx=dx/steps;

dy=dy/steps;

x=x0;

y=y0;

i=1;

while(i<=steps){

putpixel(x,y,clr);

x+=dx;

y+=dy;

i=i+1;

}

}

main(){

initwindow(800,800);

// rest

Dda\_function(200, 200, 400, 150, 1); //top

Dda\_function(200, 200, 200, 400, 1); //l line

Dda\_function(400, 150, 400, 350, 1); //r line

Dda\_function(200, 400, 400, 350, 1); //join

// seat

Dda\_function(200, 400, 350, 450, 1);

Dda\_function(400, 350, 550, 400, 1);

Dda\_function(350, 450, 550, 400, 1);

// legs

Dda\_function(200, 400, 200, 500, 1);

Dda\_function(210, 410, 210, 510, 1);

Dda\_function(200, 500, 210, 510, 1);

Dda\_function(400, 440, 400, 520, 1);

Dda\_function(410, 440, 410, 520, 1);

Dda\_function(400, 520, 410, 520, 1);

Dda\_function(340, 450, 340, 590, 1);

Dda\_function(350, 450, 350, 590, 1);

Dda\_function(340, 590, 350, 590, 1);

Dda\_function(550, 400, 550, 500, 1);

Dda\_function(540, 400, 540, 500, 1);

Dda\_function(540, 500, 550, 500, 1);

//c

Dda\_function(500, 200, 500, 300, 3);

Dda\_function(500, 200, 610, 200, 3);

Dda\_function(500, 300, 610, 300, 3);

Dda\_function(610, 280, 610, 300, 3);

Dda\_function(610, 200, 610, 220, 3);

Dda\_function(520, 220, 610, 220, 3);

Dda\_function(520, 280, 610, 280, 3);

Dda\_function(520, 220, 520, 280, 3);

getch();

closegraph();

}

# **OUTPUT**

