## Program:

#include<stdio.h>

#include<conio.h>

#include<graphics.h>

#include<dos.h>

#include<math.h>

void dda (int, int, int, int);

void main()

{

int gd = DETECT,gm;

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

dda (100,100,200,100);

dda (200,100,200,200);

dda (200,200,100,200);

dda (100,200,100,100);

dda (100,100,200,200);

dda (100,200,200,100);

dda (100,125,200,125);

dda (100,150,200,150);

dda (100,175,200,175);

dda (175,100,175,200);

dda (150,100,150,200);

dda(125,100,125,200);

dda(100,150,150,100);

dda(150,100,200,150);

dda(200,150,150,200);

dda(150,200,100,150);

getch();

closegraph();

}

void dda(int x1,int y1, int x2, int y2)

{

int i,stepsize, dx,dy;

float x,y,xinc, yinc;

dx=x2-x1;

dy-y2-y1;

x=x1;

y=y1;

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

{

stepsize=abs(dy);

}

else

{

stepsize=abs(dx);

}

xinc=dx/ (float) stepsize;

yinc=dy/ (float) stepsize;

putpixel(x,y,RED);

for(i=0; i <stepsize;i++)

{

x =x+xinc;

y=y+yinc;

putpixel((int)(x+0.5), (int)(y+0.5),RED);

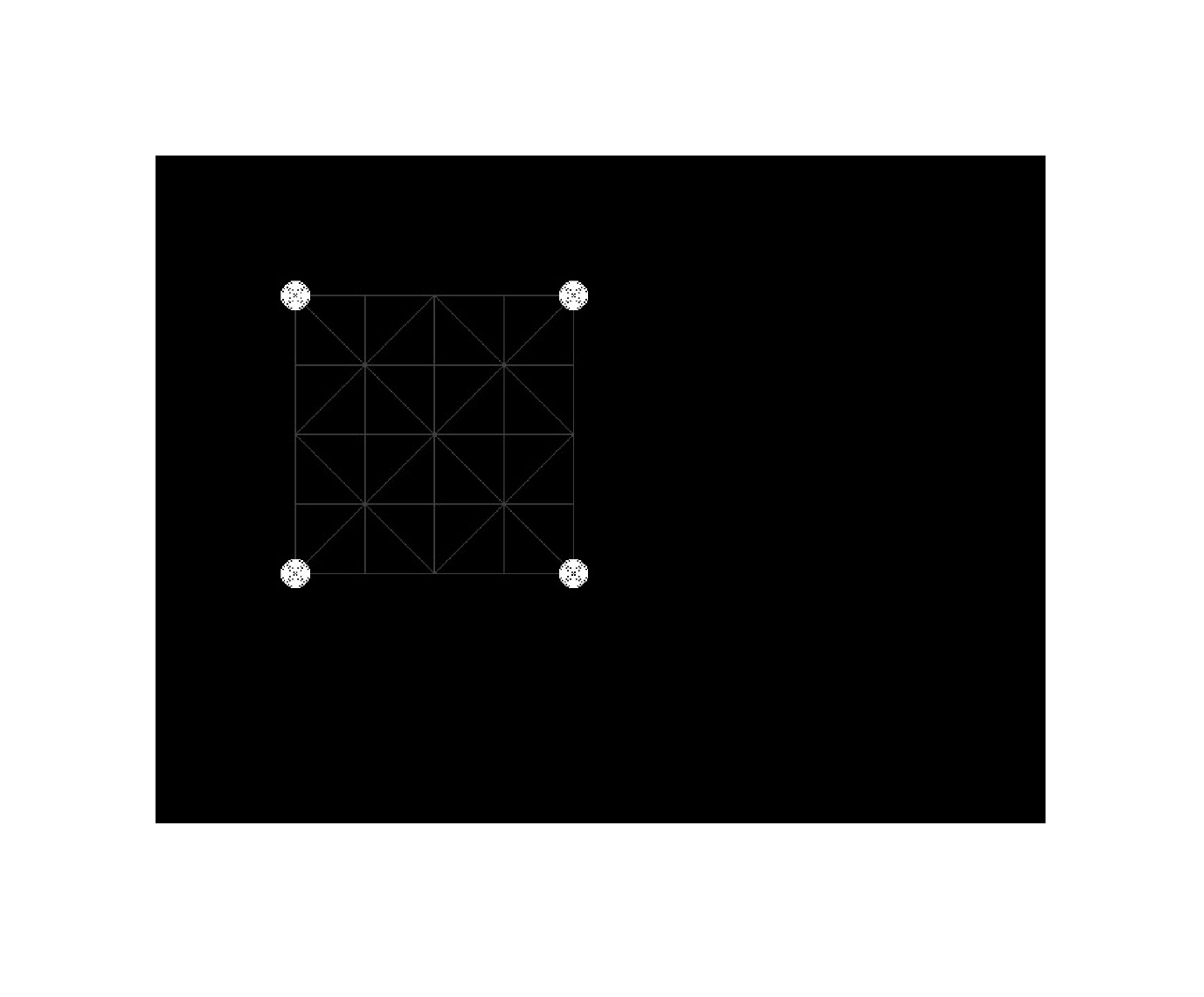
delay(10);

}

}

## Output:

The program initializes graphics mode and uses the DDA algorithm to draw a square with vertices at (100, 100), (200, 100), (200, 200), and (100, 200). It then adds the diagonals and several horizontal and vertical lines inside the square, creating a detailed grid pattern with intersecting lines, all drawn with a slight delay to animate the drawing process.



## Discussion:

In this lab, we used DDA (Digital Differential Analyzer) algorithm to produce required shape as an output. We Created a function called dda and passed the end-points as an argument and then drew a straight line from one end point to another end point

## Conclusion:

Hence, we plotted multiple straight line using DDA algorithm which as a whole look like our desired image.