DDA Algorithm:

Step1: Start Algorithm

Step2: Declare x_1,y_1,x_2,y_2,dx,dy,x,y as integer variables.

Step3: Enter value of x_1,y_1,x_2,y_2.

Step4: Calculate dx = x_2-x_1

Step5: Calculate dy = y_2-y_1

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

Step7: x_{inc}=dx/step
y_{inc}=dy/step
assign x = x_1
assign y = y_1

Step8: Set pixel (x, y)

Step9: x = x + x_{inc}
y = y + y_{inc}
Set pixels (Round (x), Round (y))

Step10: Repeat step 9 until x = x_2

Step11: End Algorithm

Example: If a line is drawn from (2, 3) to (6, 15) with use of DDA. How many points will needed to generate such line?

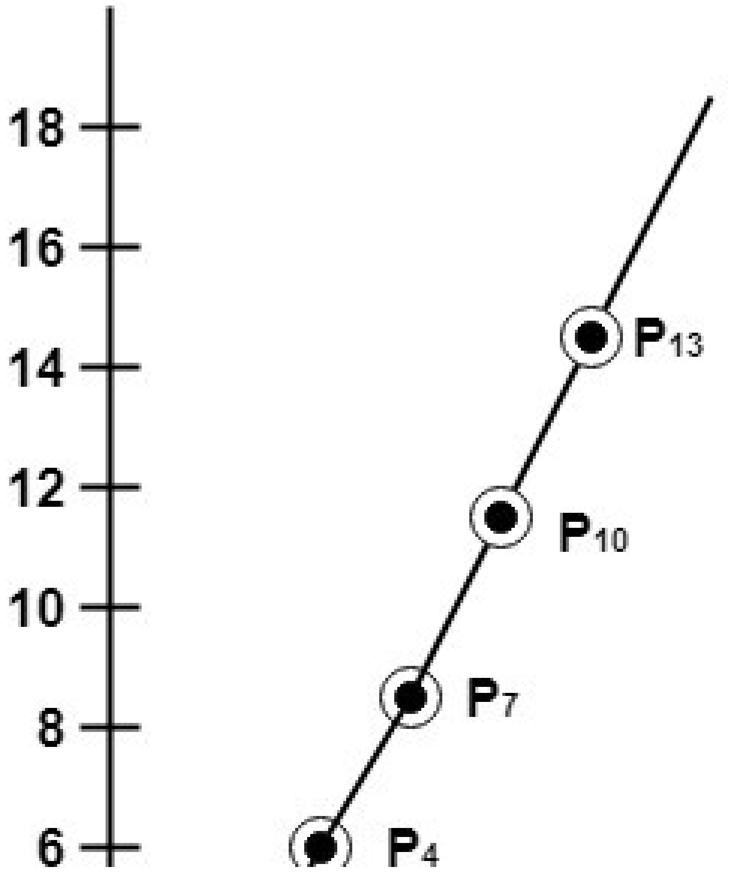
Solution: P_1 (2,3) P_{11} (6,15)

x_1=2
y_1=3
x_2= 6
y_2=15
dx = 6 - 2 = 4
dy = 15 - 3 = 12

m = $\frac{dy}{dx} = \frac{12}{4}$

For calculating next value of x takes x = x + $\frac{1}{m}$

$P_1(2, 3)$	point plotted
$P_2(2\frac{1}{3}, 4)$	point plotted
$P_3(2\frac{2}{3}, 5)$	point not plotted
$P_4(3, 6)$	point plotted
$P_5(3\frac{1}{3}, 7)$	point not plotted
$P_6(3\frac{2}{3}, 8)$	point not plotted
$P_7(4, 9)$	point plotted
$P_8(4\frac{1}{3}, 10)$	point not plotted
$P_9(4\frac{2}{3}, 11)$	point not plotted
$P_{10}(5, 12)$	point plotted
$P_{11}(5\frac{1}{3}, 13)$	point not plotted
$P_{12}(5\frac{2}{3}, 14)$	point not plotted
$P_{13}(6, 15)$	point plotted

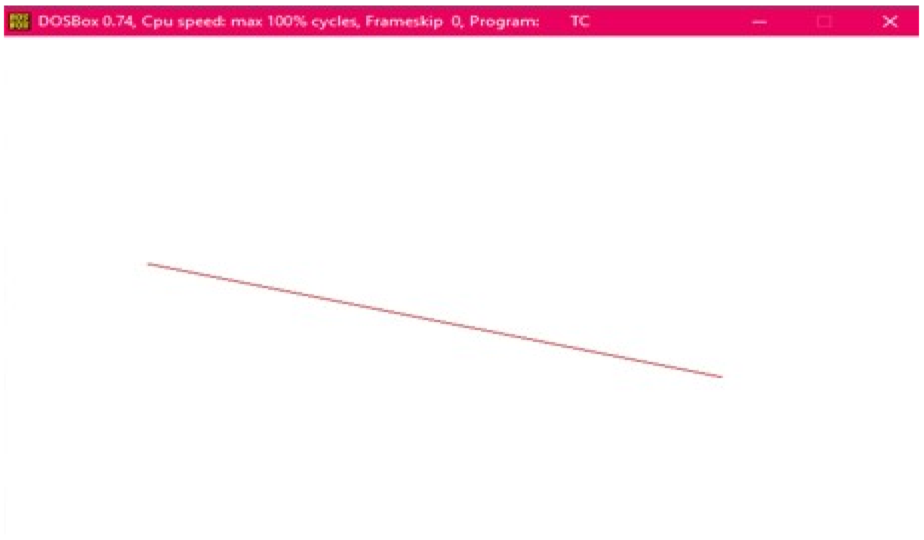


Program to implement DDA Line Drawing Algorithm:

```
#include<graphics.h>
#include<conio.h>
#include<stdio.h>
void main()
{
   intgd = DETECT ,gm, i;
   float x, y,dx,dy,steps;
   int x0, x1, y0, y1;
   initgraph(&gd, &gm, "C:\\TC\\BGI");
   setbkcolor(WHITE);
   x0 = 100 , y0 = 200, x1 = 500, y1 = 300;
   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;
    }
    getch();
    closegraph();
}
```

Output:


DOSBox 0.74, Cpu speed: max 100% cycles, Frameskip 0, Program: TC