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

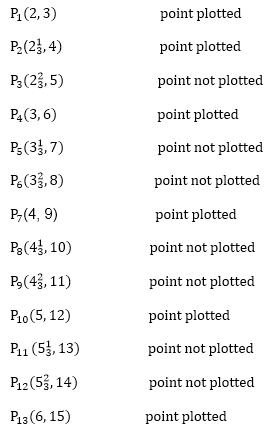
**Step11:** End Algorithm

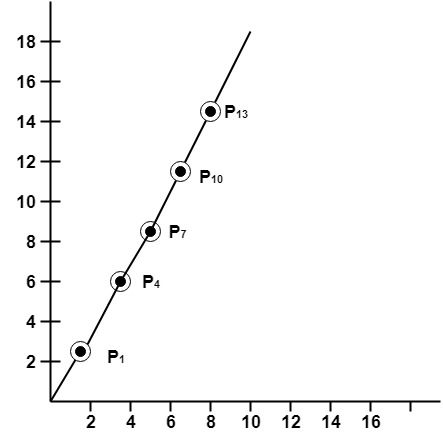
**Example:** If a line is drawn from (2, 3) to (6, 15) with use of DDA. How many points will needed to generate such line?

**Solution:** P1 (2,3)       P11 (6,15)

                x1=2  
                y1=3  
                x2= 6  
                y2=15  
                dx = 6 - 2 = 4  
                dy = 15 - 3 = 12  
                m = DDA Algorithm

For calculating next value of x takes x = x + DDA Algorithm





Program to implement DDA Line Drawing Algorithm:

1. #include<graphics.h>
2. #include<conio.h>
3. #include<stdio.h>
4. **void** main()
5. {
6. intgd = DETECT ,gm, i;
7. **float** x, y,dx,dy,steps;
8. **int** x0, x1, y0, y1;
9. initgraph(&gd, &gm, "C:\\TC\\BGI");
10. setbkcolor(WHITE);
11. x0 = 100 , y0 = 200, x1 = 500, y1 = 300;
12. dx = (**float**)(x1 - x0);
13. dy = (**float**)(y1 - y0);
14. **if**(dx>=dy)
15. {
16. steps = dx;
17. }
18. **else**
19. {
20. steps = dy;
21. }
22. dx = dx/steps;
23. dy = dy/steps;
24. x = x0;
25. y = y0;
26. i = 1;
27. **while**(i<= steps)
28. {
29. putpixel(x, y, RED);
30. x += dx;
31. y += dy;
32. i=i+1;
33. }
34. getch();
35. closegraph();
36. }

**Output:**

