Bresenham's Line Algorithm:

**Step1:** Start Algorithm

**Step2:** Declare variable x1,x2,y1,y2,d,i1,i2,dx,dy

**Step3:** Enter value of x1,y1,x2,y2  
                Where x1,y1are coordinates of starting point  
                And x2,y2 are coordinates of Ending point

**Step4:** Calculate dx = x2-x1  
                Calculate dy = y2-y1  
                Calculate i1=2\*dy  
                Calculate i2=2\*(dy-dx)  
                Calculate d=i1-dx

**Step5:** Consider (x, y) as starting point and xendas maximum possible value of x.  
                If dx < 0  
                        Then x = x2  
                        y = y2  
                          xend=x1  
                If dx > 0  
                    Then x = x1  
                y = y1  
                        xend=x2

**Step6:** Generate point at (x,y)coordinates.

**Step7:** Check if whole line is generated.  
                If x > = xend  
                Stop.

**Step8:** Calculate co-ordinates of the next pixel  
                If d < 0  
                    Then d = d + i1  
                If d ≥ 0  
          Then d = d + i2  
                Increment y = y + 1

**Step9:** Increment x = x + 1

**Step10:** Draw a point of latest (x, y) coordinates

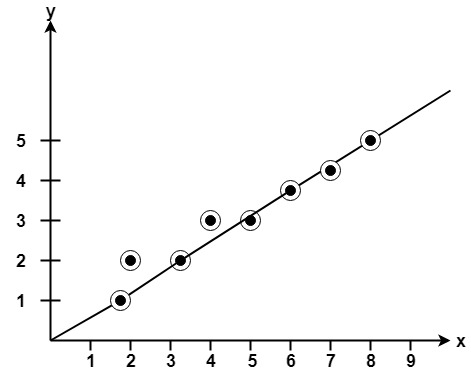
**Step11:** Go to step 7

**Step12:** End of Algorithm

**Example:** Starting and Ending position of the line are (1, 1) and (8, 5). Find intermediate points.

**Solution:** x1=1  
                y1=1  
                x2=8  
                y2=5  
                dx= x2-x1=8-1=7  
                dy=y2-y1=5-1=4  
                I1=2\* ∆y=2\*4=8  
                I2=2\*(∆y-∆x)=2\*(4-7)=-6  
                d = I1-∆x=8-7=1

|  |  |  |
| --- | --- | --- |
| **x** | **y** | **d=d+I1 or I2** |
| 1 | 1 | d+I2=1+(-6)=-5 |
| 2 | 2 | d+I1=-5+8=3 |
| 3 | 2 | d+I2=3+(-6)=-3 |
| 4 | 3 | d+I1=-3+8=5 |
| 5 | 3 | d+I2=5+(-6)=-1 |
| 6 | 4 | d+I1=-1+8=7 |
| 7 | 4 | d+I2=7+(-6)=1 |
| 8 | 5 |  |



Program to implement Bresenham's Line Drawing Algorithm:

1. #include<stdio.h>
2. #include<graphics.h>
3. **void** drawline(**int** x0, **int** y0, **int** x1, **int** y1)
4. {
5. **int** dx, dy, p, x, y;
6. dx=x1-x0;
7. dy=y1-y0;
8. x=x0;
9. y=y0;
10. p=2\*dy-dx;
11. **while**(x<x1)
12. {
13. **if**(p>=0)
14. {
15. putpixel(x,y,7);
16. y=y+1;
17. p=p+2\*dy-2\*dx;
18. }
19. **else**
20. {
21. putpixel(x,y,7);
22. p=p+2\*dy;}
23. x=x+1;
24. }
25. }
26. **int** main()
27. {
28. **int** gdriver=DETECT, gmode, error, x0, y0, x1, y1;
29. initgraph(&gdriver, &gmode, "c:\\turboc3\\bgi");
30. printf("Enter co-ordinates of first point: ");
31. scanf("%d%d", &x0, &y0);
32. printf("Enter co-ordinates of second point: ");
33. scanf("%d%d", &x1, &y1);
34. drawline(x0, y0, x1, y1);
35. **return** 0;
36. }

**Output:**

