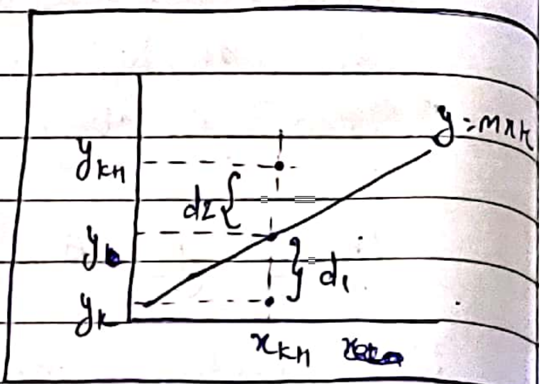
④ # Bresenham's Line Drawing Algorithm :-

This algorithm is used for scan converting a line. It is an efficient method because it involves only integers arithmatic. These operations can be performed very rapidly so lines can be generated quickly.



Slope of line is

$$y = mx + c$$

At $x_k, y_k = mx_k + c$

$x_{k+1}, y_{k+1} = mx_{k+1} + c$

When slope lies b/w

$$0 < m < 1$$

$$0 < dy/dx < 1$$

$$\therefore dy < dx$$

$d_1 = y - y_k = m(x_k+1) + c - y_k$

$d_2 = y_{k+1} - y = y_k+1 - (m(x_k+1) + c)$

$d_1 - d_2 = mx_k + m + c - y_k - y_k - 1 + mx_k + m + c$

$= 2m(x_k+1) - 2y_k + 2c - 1$

$dx(d_1 - d_2) = 2dy(x_k+1) - 2dxy_k + 2dxc - 2dx$

Decision Parameter

$p_k = 2dy\,x_k - 2dx\,y_k + d$

where $d = 2dy + dx(2c-1)$

If ($p_k < 0$), then select lower pixel

$\therefore x_{k+1} = x_k + 1$ & $y_{k+1} = y_k$

$p_{k+1} = 2dy(x_k+1) - 2dxy_k + d$

$= 2dy\,x_k + 2dy - 2dxy_k + d$

$p_{k+1} = p_k + 2dy$

Else

$x_{k+1} = x_k + 1$ & $y_{k+1} = y_k + 1$

$p_{k+1} = 2dy(x_k + 1) - 2dx(y_k + 1) + d$

$= 2dy\, x_k + 2dy - 2dx\, y_k - 2dx + d$

$p_{k+1} = p_k + 2(dy - dx)$

Now, At starting step (x_0, y_0)

$p_0 = 2dy\, x_0 - 2dx\, y_0 + 2dy + dx(2y_0 - 2\, dy/dx\, x_0 - 1)$

$= 2dy\, x_0 - 2dx\, y_0 + 2dy + 2dx\, y_0 - 2dy\, x_0 - dx$

$p_0 = 2dy - dx$

Algorithm :

i) Input end points of line $A(x_1, y_1)$ & $B(x_2, y_2)$

ii) Compute $dx = x_2 - x_1$ & $dy = y_2 - y_1$

$p = 2dy - dx$

iii) Initialize $x = x_1$ & $y = y_1$

iv) Put Pixel (x, y)

v) while $(x < x_2)$

    if $(p < 0)$ then

        $p = p + 2dy$

    else

        $p = p + 2(dy - dx)$

        $y = y + 1$

    end if

    $x = x + 1$

    Putpixel (x, y)

end while

vi) Exit.