approximate the line length

if abs($x_2 - x_1$) $\geq$ abs($y_2 - y_1$) **then**

    Length = abs($x_2 - x_1$)

else

    Length = abs($y_2 - y_1$)

end if

select the larger of Δx *or* Δy *to be one raster unit*

$\Delta x = (x_2 - x_1)/\text{Length}$

$\Delta y = (y_2 - y_1)/\text{Length}$

$x = x_1 + 0.5$

$y = y_1 + 0.5$

begin main loop

i = 1

while (i $\leq$ Length)

    setpixel (**Integer**(x), **Integer**(y))

    $x = x + \Delta x$

    $y = y + \Delta y$

    i = i + 1

end while

Line from (0, 0) to (5, 5)

$x_1 = 0, y_1 = 0, x_2 = 5, y_2 = 5$, $\boxed{\text{Length} = 5}$

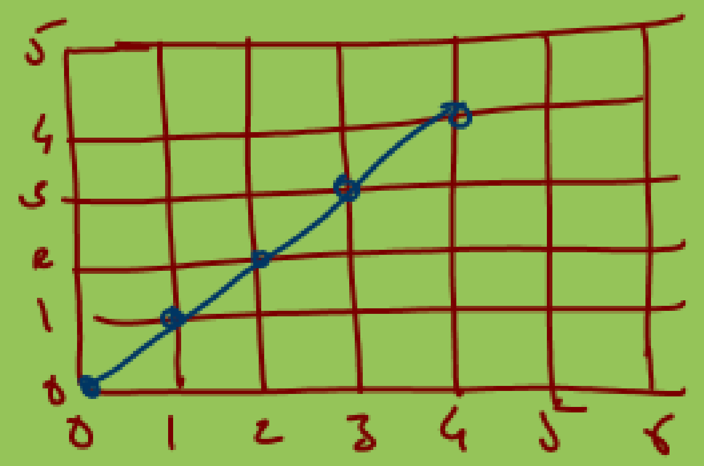
$\Delta x = (x_2 - x_1) / \text{Len} = 5/5 = 1$

$\Delta y = (y_2 - y_1) / \text{Len} = 5/5 = 1$

$x = 0.5$, $y = 0.5$

i	setpixel	x	y
		0.5	0.5
1	(0, 0)	1.5	1.5
2	(1, 1)	2.5	2.5
3	(2, 2)	3.5	3.5
4	(3, 3)	4.5	4.5
5	(4, 4)	5.5	5.5

stop



Bresenham's line rasterization algorithm for the first octant

the line end points are (x_1, y_1) *and* (x_2, y_2), *assumed not equal*

Integer *is the integer function*

x, y, Δx, Δy *are assumed integer;* e *is real*

initialize variables

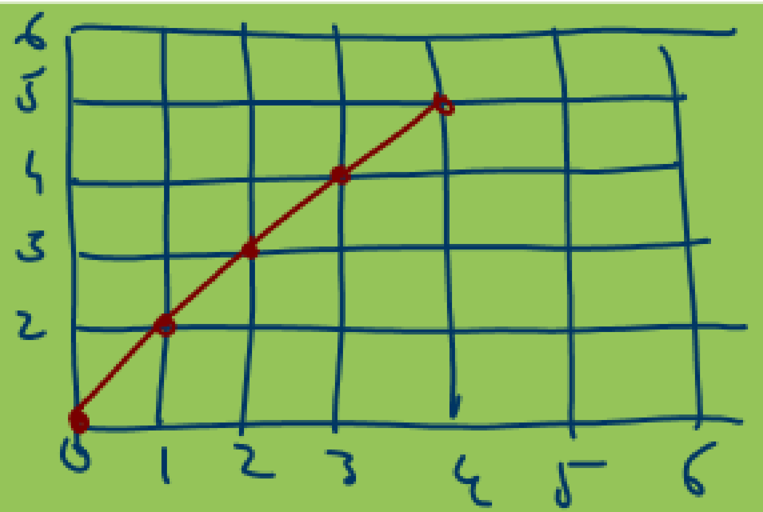
$x = x_1$
$y = y_1$
$\Delta x = x_2 - x_1$
$\Delta y = y_2 - y_1$
$m = \Delta y/\Delta x$

initialize e *to compensate for a nonzero intercept*

$e = m - 1/2$

begin the main loop

for $i = 1$ **to** Δx 5
  setpixel (x, y)
  while $(e > 0)$
    $y = y + 1$
    $e = e - 1$
  end while
  $x = x + 1$
  $e = e + m$
next i



Consider a line $(0,0)$ to $(5,5)$.
Initial calculations are
$x=0$, $y=0$, $\Delta x=5$, $\Delta y=5$, $m=1$
$e = m - 1/2 = 1 - 1/2 = +1/2$

i	setpixel	e	x	y
		$1/2$	0	0
1	(0,0)			
		$-1/2$	0	1
		$1/2$	1	1
2	(1,1)			
		$-1/2$	1	2
		$1/2$	2	2
3	(2,2)			
		$-1/2$	2	3
4	(3,3)	$1/2$	3	3
		$-1/2$	3	4
5	(4,4)	$1/2$	4	4

i	e	x	y
	$-1/2$	4	5
	$1/2$	5	5

$6 > \Delta x$ stop

Bresenham's integer algorithm *for the first octant*

the line end points are (x_1, y_1) *and* (x_2, y_2)*, assumed no*

all variables are assumed integer

initialize variables

$x = x_1$

$y = y_1$

$\Delta x = x_2 - x_1$

$\Delta y = y_2 - y_1$

initialize $\bar{e}$ *to compensate for a nonzero intercept*

$\bar{e} = 2 * \Delta y - \Delta x$

begin the main loop

```
for i = 1 to Δx
    setpixel (x, y)
    while (ē > 0)
        y = y + 1
        ē = ē − 2 * Δx
    end while
    x = x + 1
    ē = ē + 2 * Δy
next i
finish
```

Draw a line (5, 8) & (9,

Bresenham's integer algorithm *for the first octant*

the line end points are (x_1, y_1) *and* (x_2, y_2)*, assumed no*

all variables are assumed integer

initialize variables

$x = x_1$

$y = y_1$

$\Delta x = x_2 - x_1$

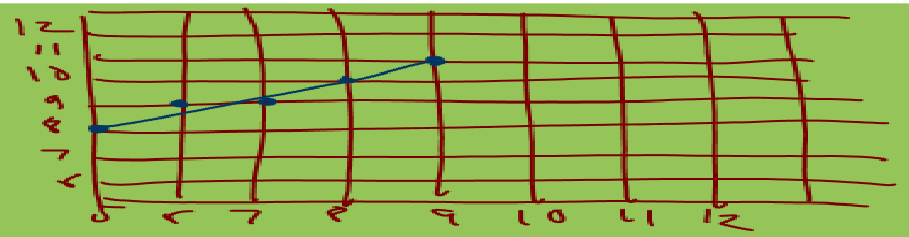
$\Delta y = y_2 - y_1$

initialize $\bar{e}$ *to compensate for a nonzero intercept*

$\bar{e} = 2 * \Delta y - \Delta x$

begin the main loop

```
for i = 1 to Δx
    setpixel (x, y)
    while (ē > 0)
        y = y + 1
        ē = ē − 2 * Δx
    end while
    x = x + 1
    ē = ē + 2 * Δy
next i
finish
```

Line (5, 8) to (9, 11)

Initial calculations

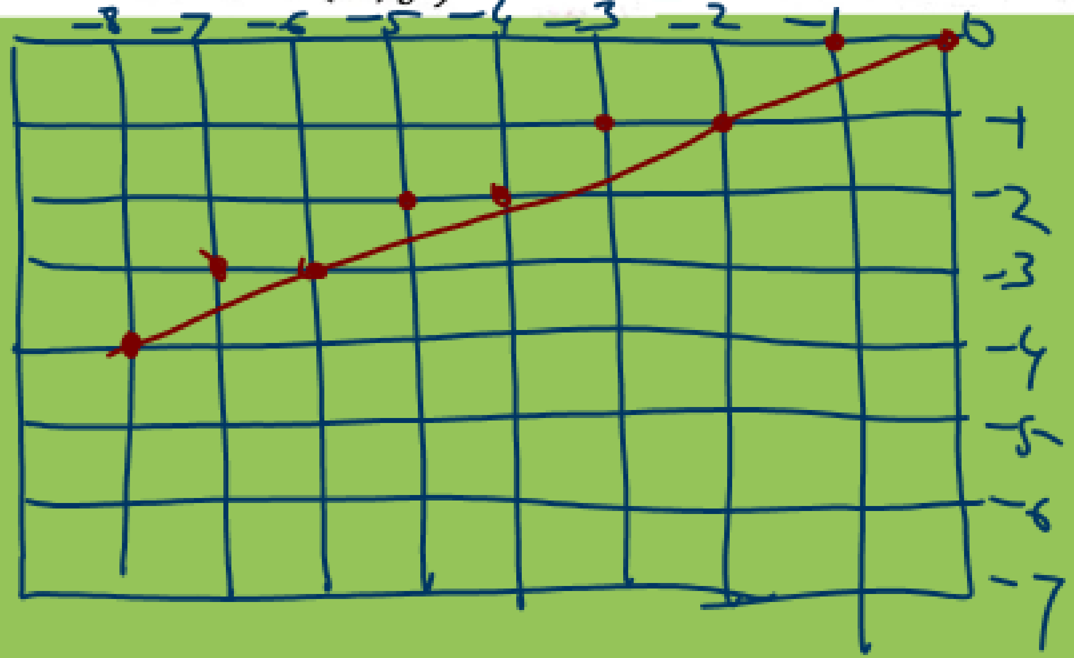
$x = 5$, $y = 8$, $\Delta x = x_2 - x_1 = 4$, $\Delta y = y_2 - y_1 = 3$

$\bar{e} = 2\Delta y - \Delta x = 2$

$2\Delta y = 6$, $2\Delta x = 8$

i	setpixel	$\bar{e}$	x	y
		2	5	8
1	(5, 8)			
		−6	5	9
		0	6	9
2	(6, 9)	6	7	9
3	(7, 9)			
		−2	7	10
4	(8, 10)	4	8	10
		−4	8	11
		2	9	11
5	(9, 11)			

```
initialize variables
x = x1
y = y1
Δx = abs(x2 − x1)
Δy = abs(y2 − y1)
s1 = Sign(x2 − x1)
s2 = Sign(y2 − y1)
interchange Δx and Δy,
if Δy > Δx then
    Temp = Δx
    Δx = Δy
    Δy = Temp
    Interchange = 1
else
    Interchange = 0
end if
initialize the error term
ē = 2 * Δy − Δx
main loop
for i = 1 to Δx
    setpixel (x, y)
    while (ē > 0)
        if Interchange = 1 then
            x = x + s1
        else
            y = y + s2
        end if
        ē = ē − 2 * Δx
    end while
    if Interchange = 1 then
        y = y + s2
    else
        x = x + s1
    end if
    ē = ē + 2 * Δy
next i
finish
```



line (0,0) to (-8,-4)

Initial calculations:

$x=0, y=0, \Delta x = 8, \Delta y = 4, s_1 = -1, s_2 = -1$

$\Delta y < \Delta x$, so Interchange $= 0$

$\bar{e} = 2\Delta y - \Delta x = 8 - 8 = 0$

i	setpixel	$\bar{e}$	x	y
		0	0	0
1	(0,0)	8	-1	0
2	(-1,0)	-8	-1	-1
		0	-2	-1
3	(-2,-1)	8	-3	-1
4	(-3,-1)	-8	-3	-2
		0	-4	-2
5	(-4,-2)	8	-5	-2
6	(-5,-2)	-8	-5	-3
		0	-6	-3
7	(-6,-3)	8	-7	-3
8	(-7,-3)	-8	-7	-4
		0	-8	-4

↑ (-8,-4)

→ Δx stop