0.0.1 render_point

"render_point(x0,y0,x1,y1,X)" is used to find the Y value at point X along the line specified by x0, x1, y0 and y1. This function uses an integer algorithm to solve for the point directly without calculating intervening values along the line.

```
1)  [dy] = [y1] - [y0]
2) [adx] = [x1] - [x0]
3) [ady] = absolute value of [dy]
4) [err] = [ady] * ([X] - [x0])
5) [off] = [err] / [adx] using integer division
6) if ( [dy] is less than zero ) {
     7) [Y] = [y0] - [off]
   } else {
     8) [Y] = [y0] + [off]
   }
9) done
```

*** Using algorithmicx, the above look as follows: ***

Algorithm 1 render_point

function RENDER_POINT(x_0, y_0, x_1, y_1, X)
    $dy \leftarrow y_1 - y_0$
    $adx \leftarrow x_1 - x_0$
    $ady \leftarrow |dy|$
    $err \leftarrow ady \cdot (X - x_0)$
    $off \leftarrow err/adx$ ▷ using integer division
    if $dy < 0$ **then**
        $Y \leftarrow y_0 - off$
    else
        $Y \leftarrow y_0 + off$
    end if
    return Y
end function

*** actually, while I am at it, is there any reason you don't use this instead: ***

Algorithm 2 render_point

function RENDER_POINT(x_0, y_0, x_1, y_1, X)
    $dy \leftarrow y_1 - y_0$
    $dx \leftarrow x_1 - x_0$
    $err \leftarrow dy \cdot (X - x_0)$
    $off \leftarrow err/dx$ ▷ using integer division
    $Y \leftarrow y_0 + off$
    return Y
end function