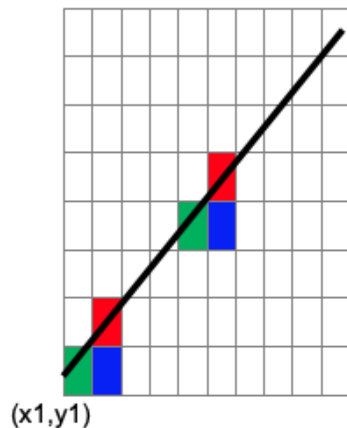


## Experiment-2

Aim: To study Bresenham Line Drawing Algorithm

## Theory

- Bresenham's Line drawing algorithm is an accurate, efficient raster line drawing algorithm, that scan converts line using only incremental integer calculations.
- Let us consider the starting point  $(x_1, y_1)$  of a given gentle slope line, we step to each successive column (x direction) and plot the pixel whose scan-line 'y' value closest to the actual line path.



- Now if we consider the next point to be plotted with respect to the point  $(x_1, y_1)$  we may select the point as either  $(x_1+1, y_1)$  or  $(x_1+1, y_1+1)$  as shown in figure 1. • For the selection of the next point Bresenham algorithm uses the decision parameter, to decide whether which point is closer to the actual line segment.
- Algorithm considers the vertical distances between the specified point  $(x_1+1, y_1)$  and  $(x_1+1, y_1+1)$  with respect to the line segment.
- As it is observed that for the gentle slope line we will be having more points on X direction, so we have to always increment in X-direction respectively.
- With the increment in X direction, the decision parameter will decide the increment in y direction as either (0/1) for the gentle slope line.
- Bresenham algorithm can work efficiently and provides more end point accuracy for the gentle slope lines as well as sharp slope lines.

## Procedure

ALGORITHM :

Let considering the line segment which has Gentle Slope  
Assume:  $x_2 > x_1$  and  $y_2 > y_1$ .

1. Start.
2. Accept end points of a line segment  $(x_1, y_1), (x_2, y_2)$ .
3. Calculate:

```
4. dx = abs(x2-x1)
5.   dy = abs(y2-y1)
```

4. Calculate the Length of the Line segment as:

```
if ( dx >= dy )
then
length = dx
else
length = dy
```

5. Calculate Initial Decision Parameter as:

```
d = 2 * dy - dx
```

6. Calculate:

```
7. incr1 = 2 * dy &ndash; 2 * dx
8.   incr2 = 2 * dy
```

7. Calculate xincr as:

```
8. If ( x2 > x1 )
9.
10. then
11.
12. yincr = 1
13.
14. else
15.
   yincr = -1
```

8. Initialize:

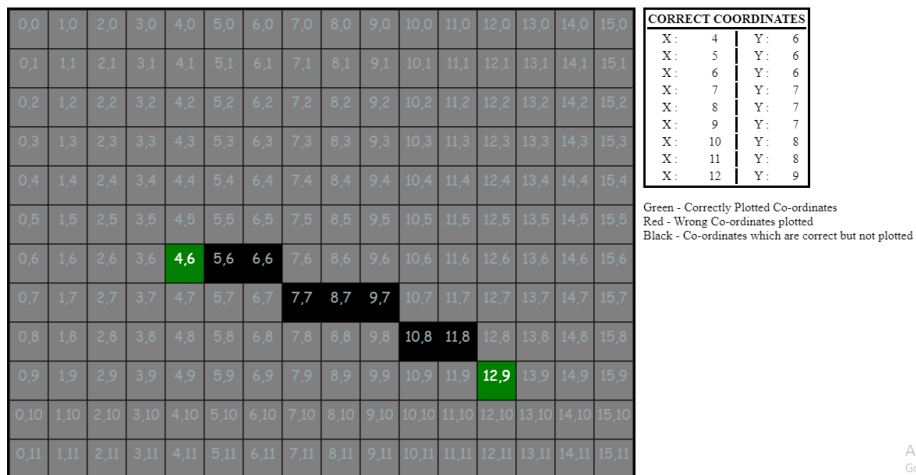
```
9. x=x1
10.
11. y=y1
12. i=1
```

13. while(i <= length)

```
14. {
15.
16.  plotpoint (x ,y)
17.
18.  if(d >= 0)
19.
20.  then
21.
22.  x = x + xincr
23.
24.  y = y + yincr
25.
26.  d = d + incr1
27.
28.  else
29.
30.  x = x + xincr
31.
32.  d = d + incr2
33.
34.  i++
35. }
```

36. Stop.

Stimulation:



Conclusion: Hence the study of Bresenham Line Drawing Algorithm is completed.