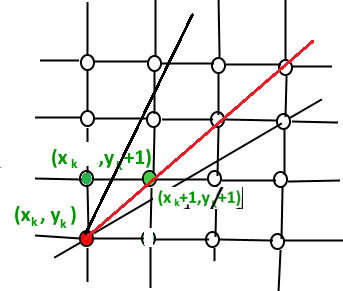
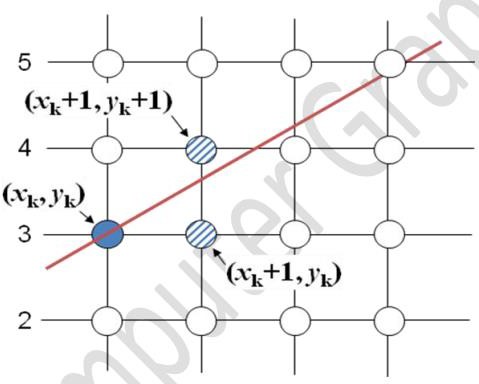
## Lab2

**Bresenham’s Line Drawing Algorithm**

**Objective:** To implement Bresenham’s line drawing algorithm for drawing a line segment between two given endpoints A (x1, y1) and B(x2, y2).

### Theory:

* Move across the x axis in unit intervals and at each step choose between two different y coordinates when slope m<1.
* For example, from position (2, 3) we have to choose between (3, 3) and (3, 4). We would like the point that is closer to the original line.
* Move across the y axis in unit intervals and at each step choose between two different x coordinates when slope is m>=1.
* So we have to take decision to choose next point. So next pixels are selected based on the value of decision parameter p. The equations are given in below algorithm.

# Bresenham’s Line Drawing Algorithm

**Step 1** : Start

**Step 2:** Input starting point P1(x1, y1) and ending pointP2 (x2,y2)

**Step 3** : Calculate the slope(m) of the required Line. **Step 4** : Identify the value of slope(m). m = dy/dx **Step 4.1** : If slope(m) is Less than 1 i.e: m < 1

* **Step 4.1.1** : Calculate the constants dx, dy,, and (2dy – 2dx) and get the first value for the decision parameter as -
* p0 = 2dy − dx
* **Step 4.1.2** : At each Xk along the line, starting at k = 0, perform the following test −
* If pk < 0, the next point to plot is (xk + 1,yk) and pk+1 = pk + 2dy

else

* plot (xk+1,yk + 1)
* pk+1 = pk + 2dy − 2dx
* **Step 4.1.3** : Repeat until x2>=x1..

**Step 4.2** : If slope(m) is greater than or equal to 1 i.e: m >= 1

* + **Step 4.2.1** : Calculate the constants dx, dy, and (2dx – 2dy) and get the first value for the decision parameter as -
  + p0 = 2dx − dy
  + **Step 4.2.2** : At each yk along the line, starting at k = 0, perform the following test −
  + If pk < 0, the next point to plot is (xk,yk + 1) and pk+1 = pk + 2dx

else

* + plot (xk + 1,yk+1)
  + pk+1 = pk + 2dx − 2dy
  + **Step 4.2.3** : Repeat until y2>=y1.

**Step 5** : Exit.

**Program:**

#include <stdio.h>

#include <graphics.h>

#include <math.h>

#include <conio.h>

int roundNow(float x){

if((x-(int)x)>=0.5) return (int)x+1;

else return (int)x;

}

void drawLine(int a, int b, int c, int d){

float x=a,y=b;

int dx=(c-a), dy=(d-b);

float m=dy/(float)dx;

if(fabs(m)<1){

while(x<=c){

putpixel(roundNow(x),roundNow(y),RED);

printf("%d, %d\n", roundNow(x), roundNow(y));

x+=1;

y+=m;

}

} else if(fabs(m)>1) {

while(y<=d){

putpixel(roundNow(x),roundNow(y),RED);

printf("%d, %d\n", roundNow(x), roundNow(y));

x+=1/m;

y+=1;

}

} else {

while(y<=d || x<=c){

putpixel(roundNow(x),roundNow(y),RED);

//printf("%d, %d\n", roundNow(x), roundNow(y));

x+=1;

y+=1;

}

}

}

int main()

{

int graphicsDriver=DETECT, graphicsMode;

int a,b,c,d;

//clrscr();

printf("Enter starting points: ");

scanf("%d %d", &a,&b);

printf("Enter ending points: ");

scanf("%d %d", &c,&d);

initgraph(&graphicsDriver, &graphicsMode, "C:\\TURBOC3\\BGI");

drawLine(a,b,c,d);

getch();

closegraph();

return 0x1337;

}